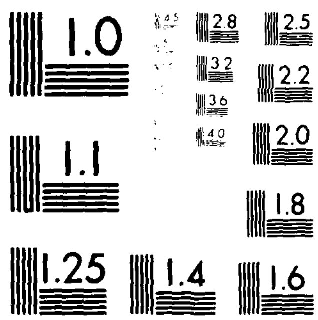
AD-A099 533 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 20/4
DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTI--ETC(U)
APR 81 A M REED
UNCLASSIFIED DTNSRDC/SPD-0927-02 NL

1 OF 3
AD A
099533

9953



MICROCOPY RESOLUTION TEST CHART

LEVEL II

12

DTNSRDC/SPD-0927-02

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20084

AD A099533

DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTION PROGRAMS (CLOSEFIT AND SYNTHESIS): MAINTENANCE MANUAL

DOCUMENTATION FOR SWATH SHIP

RESISTANCE AND PROPULSION PREDICTION PROGRAMS

(CLOSEFIT AND SYNTHESIS):

MAINTENANCE MANUAL

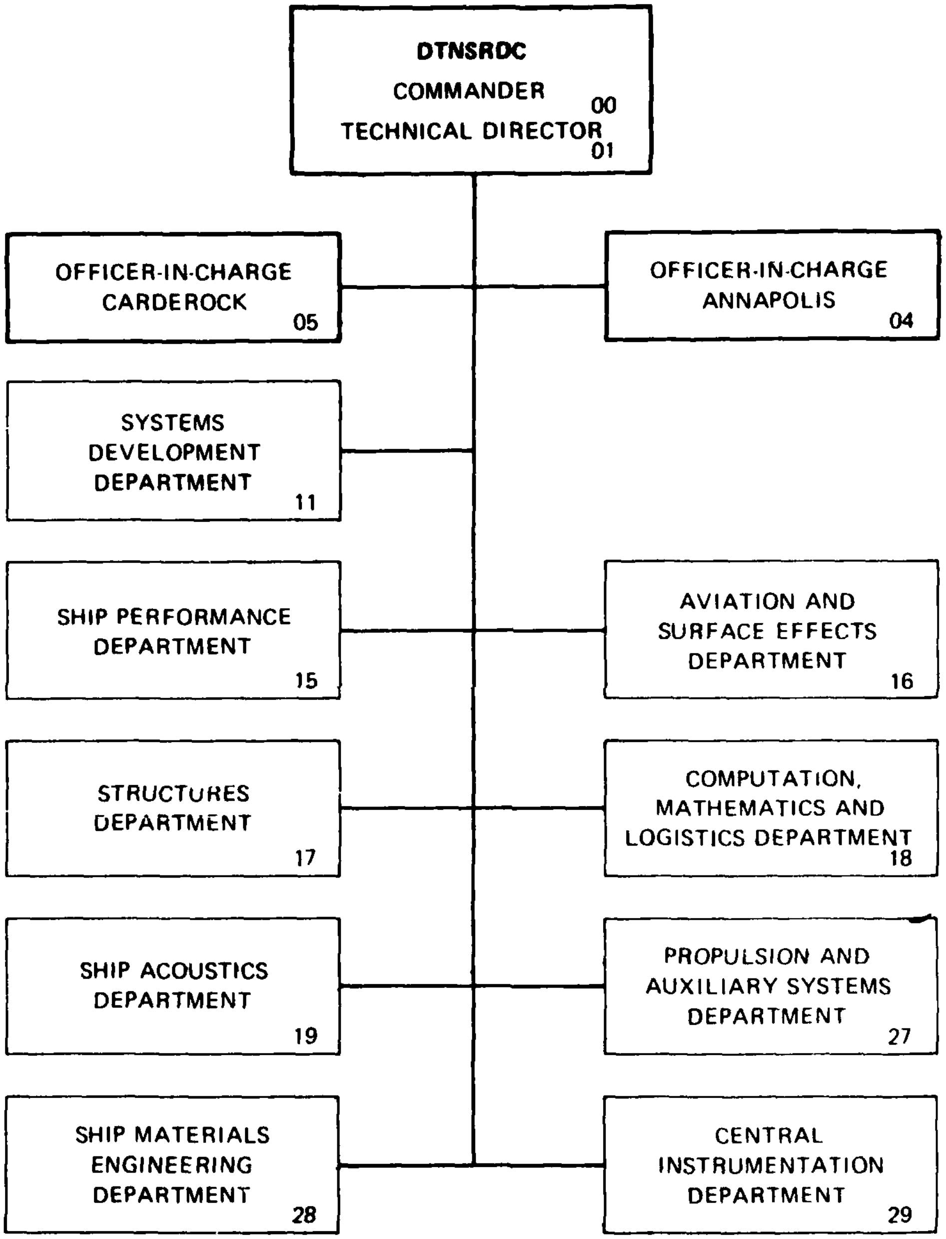
Arthur M. Reed

Approved for Public Release: Distribution Unlimited

Ship Performance Department

APRIL 1981 DTNSRDC/SPD-0927-02

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED 387 662

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: DTNSRDC/SPD-0927-02
2. GOVT ACCESSION NO.: AD-A099 533
3. RECIPIENT'S CATALOG NUMBER: 9

4. TITLE (and Subtitle): 6 DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTION PROGRAMS (CLOSEFIT AND SYNTHESIS): MAINTENANCE MANUAL.

5. TYPE OF REPORT & PERIOD COVERED: 11 FINAL rept.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): 10 Arthur M./Reed 12 232

8. CONTRACT OR GRANT NUMBER(s): 17

9. PERFORMING ORGANIZATION NAME AND ADDRESS: David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland 20084

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62543N, ZF43 421 001 1-1500-103

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE: 11 APR[redacted]81

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 16 F43421

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release: Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Small-Waterplane-Area Twin-Hull (SWATH) ships, resistance and propulsion prediction

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents two computer programs which determine the wave resistance and propulsive performance of Small-Waterplane-Area Twin-Hull (SWATH) ships. These programs are SYNTHESIS and CLOSEFIT, and they use as input data the moments and offsets of ship geometry, respectively. While they both employ linearized ship wave theory and thin-ship approximation and share many subroutines, CLOSEFIT is a more refined program capable of furnishing superior results. On the other hand, SYNTHESIS is simpler and includes several additional subroutines which can determine the characteristics and performance

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

i

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

389694

of a SWATH ship's propulsive system at both design and off-design conditions.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

Symbol	Description
$A(x)$	Body sectional area function
A_o	Maximum section area of body
A_{Bm}	Symmetric coefficients of Chebychev sine series for body
A_{Sm}	Symmetric coefficients of Chebychev sine series for strut
A_{Wp}	Area of the waterplane
b	Half of the separation distance of the hulls
B_{Bm}	Anti-symmetric coefficients of Chebychev series for body
B_{Sm}	Anti-symmetric coefficients of chebychev sine series for strut
C_A	Correlation Allowance
C_F	Frictional resistance coefficient
C_{Fm}	Form drag coefficient
C_P	Body Prismatic coefficient
C_{WP}	Waterplane coefficient
C_{IW}	Waterplane inertia coefficient
g	Acceleration due to gravity
h_B	Maximum depth of submergence of axis of body
h_S	Maximum draft of strut
$J_n(\alpha)$	Bessel functions
L_B	Maximum length of body
L_S	Maximum length of strut
P_E	Effective power

NOMENCLATURE (CONT

Symbol	Description
R_n	Reynolds number
R	Total ship resistance
R_{W_B}	Wave resistance due to one main body
R_F	Frictional resistance
R_{Fm}	Eddy resistance
R_{W_S}	Wave resistance due to one strut
$R_{W_{SB}}$	Wave resistance due to the interaction of strut and main body
R_W	Wave resistance
S	Wetted surface
T	Thickness of mid length of strut
t(x)	Strut half thickness function
T_Bmn	Auxilaiary function used in calculation of R_B
T (Tmax also used)	Thickness of mid length of strut
T_{SBmn}	Auxiliary wave resistance function used in calcuation of R_{SB}
T_{Smn}	Auxiliary wave resistance function used in calculation of R_S
Um(x)	Cheby chev cosine series
V	Velocity of ship
Vm(x)	Chebychev sin series
W_{Bmn}	Auxiliary wave resistance function used in calculation of R_B
W_{Smn}	Auxiliary wave resistance used in calculation of R_S

NONMENCLATURE (CONT)

Symbol	Description
W_{SBmn}	Auxiliary wave resistance function used in calculation of R_{SB}
	Variable of integration
ρ	Density of water
γOB	Dimensionless number related to body length and ship speed, used to remove singularities
γOS	Dimensionless number related to strut length and ship speed, used to remove singularities
ζ	Variable of integration.

ENGLISH/SI EQUIVALENTS

1 degree (angle)	=	0.01745 rad (radians)
1 foot	=	0.3048 m (meters)
1 foot per second (fps)	=	0.3048 m/sec (meters per second)
1 inch	=	25.40 mm(millimeters)
1 knot	=	0.5144 m/s (meters per second)
1 lb (force)	=	4.448 N (Newtons)
1 lb (force - inch lbs)	=	0.1130 N·m (Newton-meter)
1 long ton (2240 pounds)	=	1.016 metric tons; or 1016 kilograms
1 power	=	0.746 kW (kilowatts)

ABSTRACT

This report documents two computer programs which determine the wave resistance and propulsive performance of Small-Waterplane-Area Twin-Hull (SWATH) ships. These programs are SYNTHESIS and CLOSEFIT, and they use as input data the moments and offsets of ship geometry, respectively. While they both employ linearized ship wave theory and thin-ship approximation and share many subroutines, CLOSEFIT is a more refined program capable of furnishing superior results. On the other hand, SYNTHESIS is simpler and includes several additional subroutines which can determine the characteristics and performance of a SWATH ship's propulsive system at both design and off-design conditions.

ADMINISTRATIVE INFORMATION

This investigation was authorized under a direct funded block from the Naval Material Command (NAVMAT 08T23) under Program Element 62543N, Task Area ZF43-421-001, and funded through the High Performance Vehicles Office of the David W. Taylor Naval Ship Research and Development Center (DTNSRDC), Work Unit 1-1500-103.

BACKGROUND

This report (Maintenance Manual) contains documentation for two computer programs, CLOSEFIT and SYNTHESIS, which have been developed at the David W. Taylor Naval Ship R&D Center (DTNSRDC) to predict the effective and delivered power for Small-Waterplane-Area Twin-Hull (SWATH) ships. This maintenance manual contains documentation for each subroutine in the programs, definitions of all of the variables in the COMMON BLOCKS, and a listing of the computer programs. This documentation is intended for the individual who must modify the computer programs, and must therefore be able to understand the functioning of the programs, and should be contrasted with the Users Manual, which serves to define the input to and output from the programs. The ability to prepare proper input, and to use the output does not require an understanding of the workings of the program.

SWATH ships are catamarans which combine a thin strut with a submerged elongated slender body. Each demi-hull of the SWATH has either one (single) or two (tandem) struts which support the lower hull. Figures 1 and 2 show schematic diagrams illustrating the geometry of single and tandem strut SWATH configurations, respectively.

Programs CLOSEFIT and SYNTHESIS have a similar theoretical foundation in that they are both based on linearized ship wave theory and the thin ship approximation in the framework of the potential flow of an incompressible, inviscid fluid. In general, thin-ship theory can be applied to any conventional catamaran with arbitrary camber. However, as this theory has been specialized to SWATH configurations, the ship geometry is

restricted to ships with "no-cross flow" cambers. In practice, this means SWATHS with very small strut thicknesses compared to the transverse strut separation.

In the theory of ship resistance extrapolation, the resistance of a ship is typically decomposed into residuary and frictional resistance components. The frictional resistance component is generally taken as equal to the resistance of a flat plate with area of the ship, and with length equal to the length of the ship. The three dimensional viscous effects are then included in the residuary resistance along with other resistance components such as wave making resistance. For a SWATH ship, the calculation of the frictional resistance is not as simple as stated above, however, the same principle applies if one calculated the frictional resistance of the bodies and struts separately.

The wave making resistance component of the residuary resistance can be computed using inviscid fluid flow theory. However, the other components of the residuary resistance are not easily calculated theoretically. Because of this, the remainder of the residuary resistance is computed empirically. The difference between the experimentally determined residuary resistance and the theoretically determined wave resistance is calculated for those cases where adequate experimental data exists. These differences which are denoted as "form drag", are then plotted as a function of strut speed-length ratio. Under the assumption that the form drag remains unchanged for reasonable changes of SWATH proportions, a curve can be faired through the individual form drag curves, and applied to new hull forms.

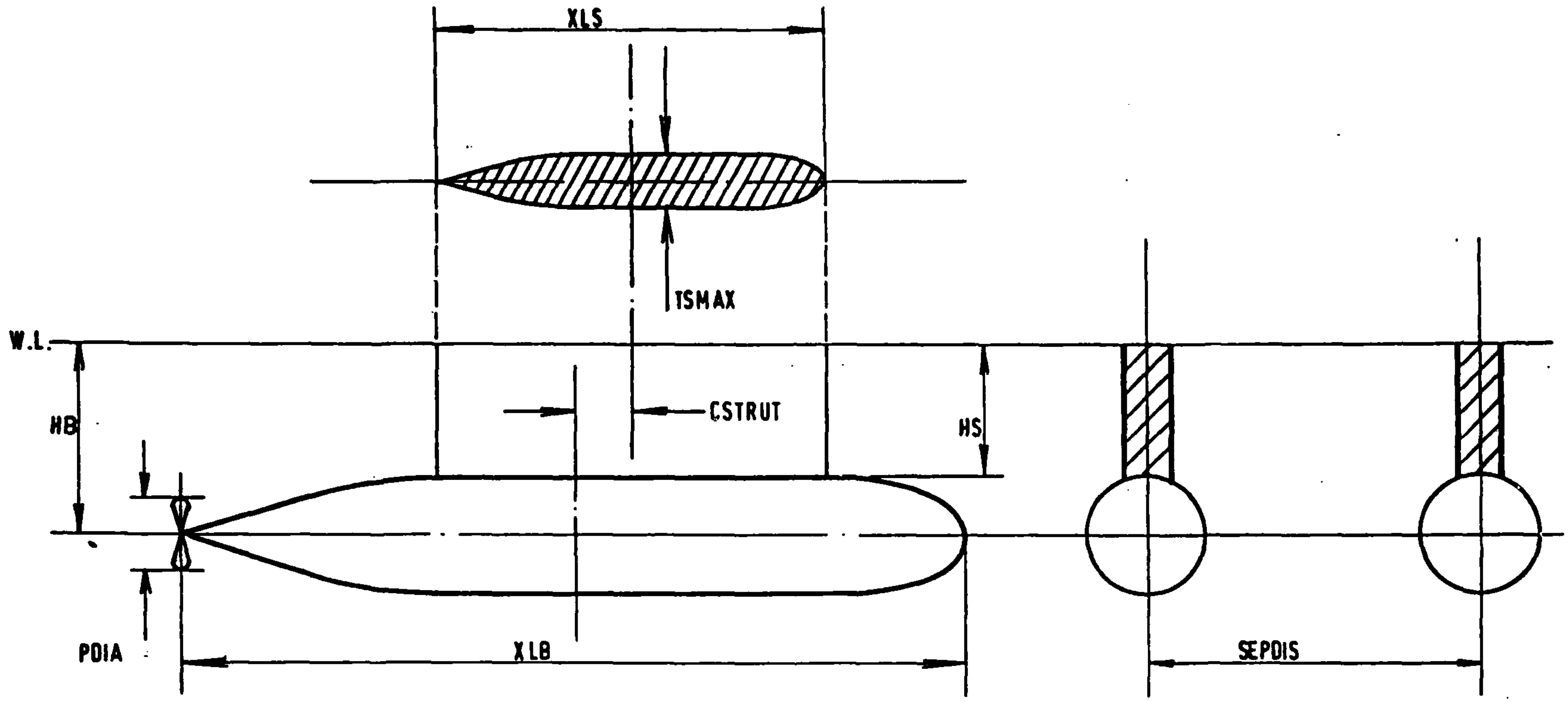


Figure 1 - Profile of SWATH Ship with Single Strut

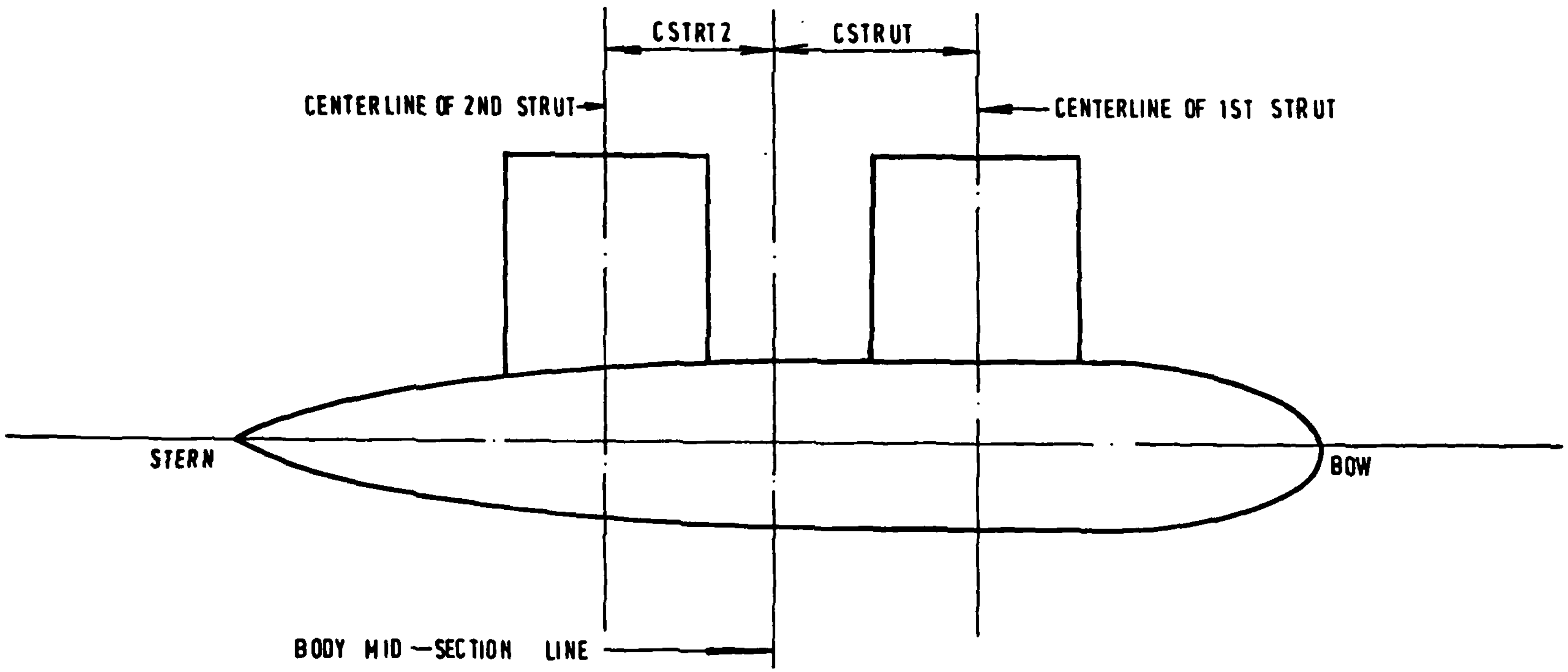


Figure 2 - Profile of SWATH Ship with Tandem Struts

For the wave resistance predictions, the geometries of the body and strut(s) are represented by modified Chebychev series. The methods used to determine these series are the principal differences between the programs CLOSEFIT and SYNTHESIS. The program CLOSEFIT uses an integration technique, similar to that used to determine the coefficients of a Fourier series, to determine the coefficients of the Chebychev series based on the actual hull offsets. The program SNYTHESIS uses statical moments of the strut waterplane and body to define the coefficients of the Chebychev series. Typically, the CLOSEFIT program uses a series of 10 to 20 pairs of terms to represent a body or strut, where each pair of terms contains an even and an odd Chebychev function, in the same fashion that a Fourier series consists of pairs of sine and cosine functions. The program SYNTHESIS uses a series consisting of only three pairs of terms to represent the body and strut. This reduction in the number of terms results in savings of ten to forty times in the computation of the wave resistance in SYNTHESIS as compared to CLOSEFIT.

In addition to the effective power computations, the program SYNTHESIS also computes the delivered power which would be required to propel the SWATH ship. The propulsion calculations are performed by a subprogram SHPCMP and its attendant subprograms. SHPCMP determines the wake and thrust deduction for the design based on empirical relations derived from earlier SWATH propulsion experiments. The program then calculated the optimum four bladed Troost series propeller at the design speed, along with the off design performance of the propeller. This data along with the effective power computations is used to calculate the delivered power.

OVERVIEW OF CLOSEFIT

The CLOSEFIT SWATH program is composed of several modules, each of which consists of a number of subroutines and performs a specific function in the SWATH ship resistance calculation. Figure 1 presents a tree diagram which reflects the relationship among all subroutines.

Subroutine CHEB evaluates the coefficients of the Chebychev series, which represents the strut half thickness function and the body sectional area function in wave resistance equations. This is accomplished by calling READ, SCALE, COMPUT, CHECK and SURFACE. The offset tables are read in by Subroutine READ, then normalized by Subroutine SCALE. The driver Subroutine COMPUT calls Subroutine CHEV, which in turn calls SPLINE and SUMSPL to compute the coefficients of the Chebychev series. The integrals used to evaluate the strut Chebychev coefficients are as follows:

$$A_{sm} = \frac{2}{\pi} \int_{-\pi/2}^{\pi/2} t(\sin\theta)\cos(2m-1)\theta d\theta$$

$$B_{sm} = \frac{2}{\pi} \int_{-\pi/2}^{\pi/2} t(\sin\theta)\sin 2m\theta d\theta$$

where $t(\sin\theta)$ is a function adopted to describe the offsets of the strut or the body. The Cubic Spline Function Technique is used to fit an analytic curve to the discrete offset points. Subroutine SPLINE solves

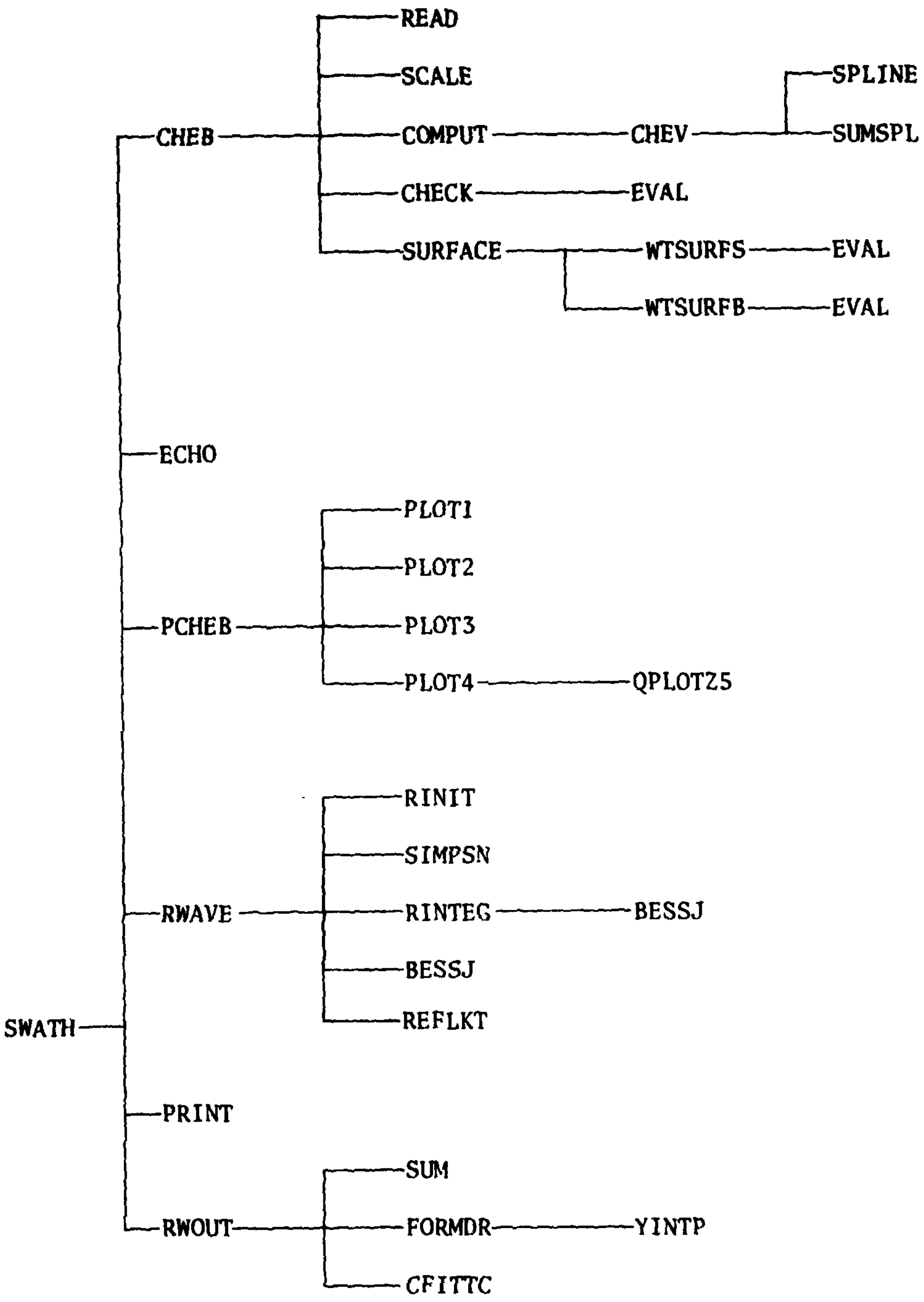


Figure 3 - Tree Diagram of CLOSEFIT SWATH Program

the set of simultaneous linear algebraic equations, which results from the spline fitting procedure and obtains the second derivatives of the analytical functions t(sin θ) at each offset point. Next, Subroutine SUMSPL uses these second derivatives in a set of integration formulas based on the Filon Quadrature Technique to evaluate the above trigonometric integrals. Subroutine CHECK evaluates the offsets approximated from the Chebychev series in Subroutine COMPUT and prints out the predicted offsets along with the true offsets. This permits a check on the accuracy of the Chebychev series approximations.

Subroutine SURFACE computes the total SWATH ship wetted surface by calling WTSURFS and WTSURFB. Subroutine ECHO echoes the input data of the SWATH program on the printer as a check of its accuracy. Subroutine PCHEB plots the strut half thickness curve and body sectional area curve on the printer in order to validate the Chebychev series approximation.

A large amount of the execution time of this program is spent in the evaluation of the auxiliary functions, T and W, given by:

$$\left\{\begin{matrix} \dfrac{T_{Smn}}{(2m-1)(2n-1)} \\[2ex] \dfrac{W_{Smn}}{(2m)(2n)} \end{matrix}\right\} = \int_{\gamma_{0S}}^{\infty} \frac{d\alpha}{\alpha^2\sqrt{\alpha^2-\gamma_{0S}^2}}\, D\left(\alpha; \frac{2b}{L_S}, \gamma_{0S}\right) \times F_S^2(\alpha) \left\{\begin{matrix} J_{2m-1}(\alpha)J_{2n-1}(\alpha) \\ J_{2m}(\alpha)\,J_{2n}(\alpha) \end{matrix}\right\},$$

$$\left\{\begin{matrix} \dfrac{T_{Bmn}}{(2m-1)(2n-1)} \\[2ex] \dfrac{W_{Bmn}}{(2m)(2n)} \end{matrix}\right\} = \int_{\gamma_{0S}}^{\infty} d\alpha\, \frac{\alpha^2}{\sqrt{\alpha^2-\gamma_{0S}^2}}\, D\left(\alpha; \frac{2b}{L_S}, \gamma_{0S}\right) \times F_B^2(\beta) \left\{\begin{matrix} J_{2m-1}(\beta)J_{2n-1}(\beta) \\ J_{2m}(\beta)J_{2n}(\beta) \end{matrix}\right\}$$

$$\left\{\begin{matrix} \dfrac{T_{SBmn}}{(2m-1)(2n-1)} \\ \dfrac{W_{SBmn}}{(2m)(2n)} \end{matrix}\right\} = \int_{\gamma_{0S}}^{\infty} \frac{d\alpha}{\sqrt{\alpha^2 - \gamma_{0S}}} D\left(\alpha; \frac{2b}{L_S}, \gamma_{0S}\right)$$

$$\times E_S(\alpha) E_B(\beta) \left\{\begin{matrix} J_{2m-1}(\alpha) J_{2n-1}(\beta) \\ J_{2m}(\alpha) J_{2n}(\beta) \end{matrix}\right\}.$$

Due to the highly oscillatory nature of the integrands of these auxiliary functions, the integration range is divided into three separate intervals. For each interval, a particular numerical integration method is used to improve the accuracy of the calculation. To avoid the singularity in the denominator of the integrand, the integration variable (α) is replaced by $Z = \sqrt{\alpha - \gamma_{OS}}$ in the interval from γ_{OS} to $\gamma_{OS} + 1$. The numerical method used in this interval is Simpson's Rule, which uses 30 points. The central region of the integration process goes from $\alpha = \gamma_{OS} + 1$ to $\alpha = \alpha_{max}$. The choice of α_{max} is based on the requirement for accuracy in the numerical integration method used in this region, where a step size is computed at a given α, and a three-point Simpson's Rule is used until the upper integration limit of the region, α_{max}, is reached.

An analytical integration formula was developed to determine the effects of the final region, from $\alpha = \alpha_{max}$ to $\alpha = \alpha_{\ell smax}$, where $\alpha_{\ell_{smax}}$ is an empirical constant which defines the upper limit of the tail region. Beyond this point, the integration area is assumed to be negligible.

Subroutine PRINT will print the table of the T and W arrays, provided the flag JTEST is set other than zero.

Subroutine RWOUT calls SUM to evaluate the components of the wave

resistance, based on Lin and Day's equations[1]. Form drag is evaluated by Subroutine FORMDR, which is based on experimental data. Frictional drag is estimated by calling Subroutine CFITTC, which is based on the C_F curve adopted by the International Towing Tank Conference (ITTC)[2]. For each test speed, a table is printed showing all the components of total ship resistance and the effective power required to tow the SWATH ship at a constant speed.

[1]References are listed on page 112.

OVERVIEW OF SYNTHESIS

The procedure of the SYNTHESIS SWATH program parallels the CLOSEFIT program in calculating total ship resistance, except in its approach to obtaining the coefficients of the Chebychev series. In SYNTHESIS, the waterplane area coefficients, waterplane moment, waterplane inertia, body prismatic and body moment are used to evaluate the Chebychev coefficients. The only other major difference between the two programs is that SYNTHESIS includes a propeller design and optimization program.

Figure 2 presents the tree diagram of all subroutines called by the SYNTHESIS program. The module CHEB is much simpler in SYNTHESIS than its CLOSEFIT counterpart due to its approach in evaluating the Chebychev coefficients through geometric identities as opposed to numerical integrations.

Modules RWAVE and ROUT are identical for both programs, and in both require a great deal of execution time.

Module FINDRG evaluates the fin drag of a SWATH ship. It is directly excerpted from coding documented in reference (3).

Module SHPCMP can determine propeller design and off-design performance if the propeller diameter is specified. The subroutines used in this module are excerpted from coding documented in reference (4).

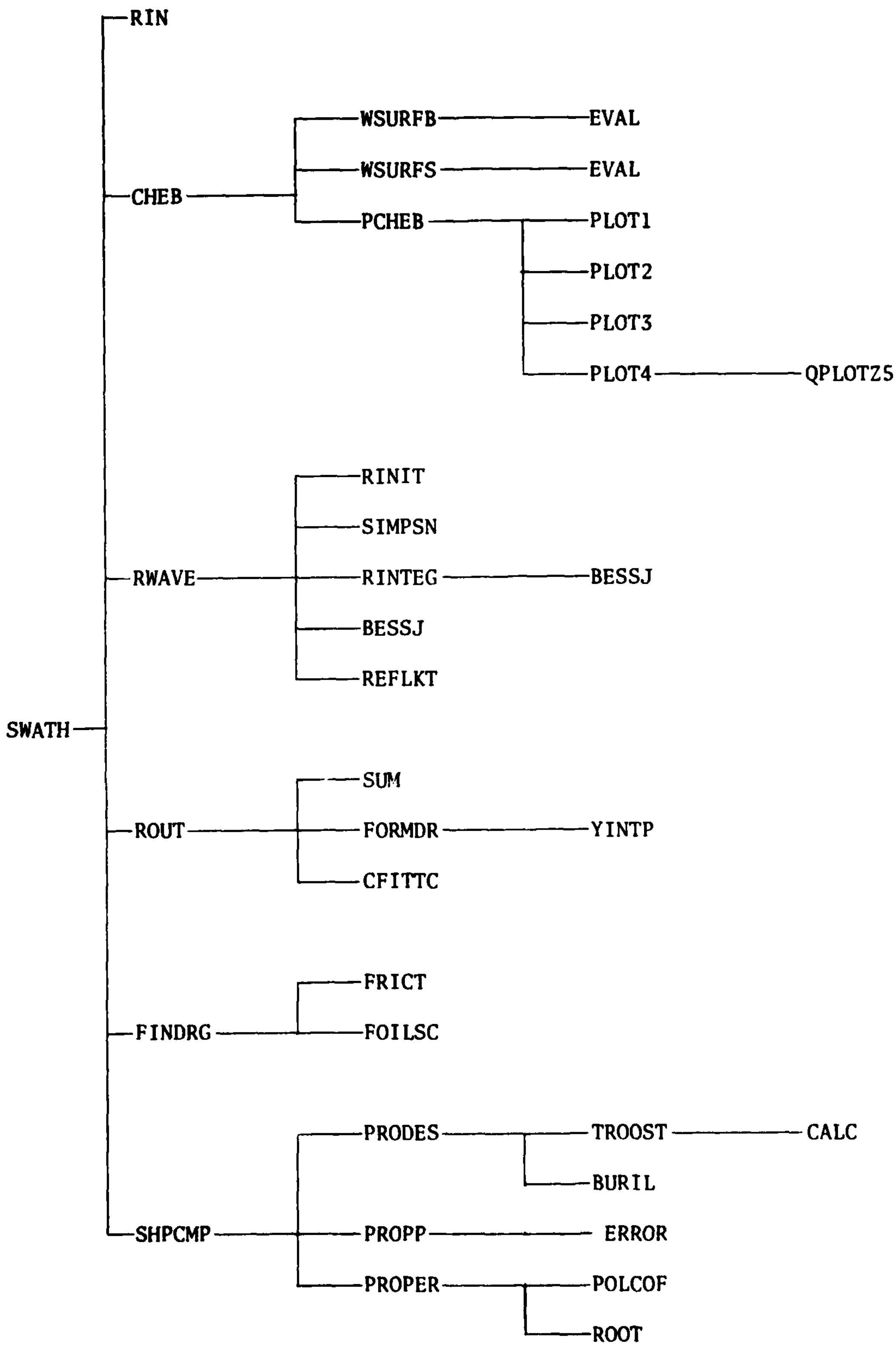


FIGURE 4 - Tree Diagram of SYNTHESIS SWATH Program

PROGRAM DOCUMENTATION OF CLOSEFIT

COMMON BLOCK DEFINITIONS FOR CLOSEFIT

COMMON/AUX/

PURPOSE: AUX stores the auxiliary function table.

NAME	TYPE	LENGTH	DEFINITION
TS	R	(10,10)	Auxiliary function T for strut
WS	R	(10,10)	Auxiliary function W for strut
TB	R	(10,10)	Auxiliary function T for body
WB	R	(10,10)	Auxiliary function W for body
TSB	R	(10,10)	Auxiliary function T for strut to body
WSB	R	(10,10)	Auxiliary function W for strut to body
TS12	R	(10,10)	Auxiliary function T for strut 1 to strut 2
WS12	R	(8,10,10)	Auxiliary function W for strut i to strut 2
TB12	R	(8,10,10)	Auxiliary function T for body 1 to body 2
WB12	R	(8,10,10)	Auxiliary function W for body 1 to body 2
TSB12	R	(8,10,10)	Auxiliary function T for strut 1 to body 2
WSB12	R	(8,10,10)	Auxiliary function W for strut 1 to body 2
TSBP	R	(10,10)	Auxiliary function T
WSBP	R	(10,10)	Auxiliary function W
TSB12P	R	(10,10)	Auxiliary function T
WSB12P	R	(10,10)	Auxiliary function W

COMMON/NAME/

PURPOSE: NAME stores a flag indicating whether the offset data are for the strut or the body and providing the alphanumeric information for the strut and the body.

NAME	TYPE	LENGTH	DEFINITION
INCHAR	I	8	INCHAR (1) = STRUT. Strut offset data will follow. INCHAR (1) = BODY. Body offset data will follow. INCHAR (2) to INCHAR (8). Hold the alphanumeric information for the strut or the body.
NAME	I	7	Temporary storage location for alphanumeric strut or body information.

COMMON/NORM/

PURPOSE: NORM stores the scaling factors for normalizing X and Y.

NAME	TYPE	LENGTH	DEFINITION
YMAX	R		Maximum value of Y
XDIVER	R		Divisor for normalizing X
YDIVER	R		Divisor for normalizing Y
XADDER	R		Number to be added to normalize X

COMMON/OMEGA/

PURPOSE: OMEGA stores the variables used in evaluating the auxiliary function.

NAME	TYPE	LENGTH	DEFINITION
NMAX	I		Maximum order of Chebychev Series
HSOLS	R		Ratio of draft of strut to length of strut (h_s/L_s)
HBOLB	R		Ratio of depth of submergence to length of body (h_b/L_b)
GAMAOS	R		$\gamma_{os} = \frac{1}{2}\, g\, L_S/V^2$
GAMAOB	R		$\gamma_{ob} = \frac{1}{2}\, g\, L_b/V^2$
NOSEPS	I		Number of different hull separation distances
SEPDIS	R	(8)	Hull separation distance b (ft)
SEP	R	(8)	$2b/(\gamma_{os} \cdot L_s)$
CSTRUT	R	(8)	Distance of CL of strut to CL of body
CSTRT2	R	(8)	Distance of CL of 2nd strut (if present) from CL of body
GOSQ	R		γ_{os}^2
PHIS	R		$2(\frac{h_s}{L_s})/\gamma_{os}$
PHIB	R		$2(\frac{h_b}{L_b})/\gamma_{ob}$
RATIOL	R		γ_{ob}/γ_{os}

COMMON/OUT/

PURPOSE: OUT defines and stores physical constants and geometric parameters.

NAME	TYPE	LENGTH	DEFINITION	SET VALUES
HS	R		Draft of strut h_s (ft)	
HB	R		Depth of submergence from surface to body centerline h_b (ft)	
XLS	R		Length of strut L_S (ft)	
XLB	R		Length of body L_b (ft)	
TSMAX	R		Maximum thickness of strut t_{max} (ft)	
AX	R		Maximum cross-sectional area of body A_X (ft^2)	
PI	R		3.1415926535897	
G	R		Acceleration due to gravity g (ft/sec^2)	32.155
RHO	R		Density of water ρ ($lb \times sec^2/ft^4$)	1.9367
GNU	R		kinematic viscosity ν (ft^2/sec)	1.2970×10^5
WETS	R		Wetted surface area of strut S_s (ft^2)	
WETB	R		Wetted surface area of body S_b (ft^2)	
WTSURF	R		Total wetted surface area S_t (ft^2)	
VMFPS	R		Test velocity V (fps)	
DELCF	R		Correction allowance ΔC_F	0.0005
TITLE	I	(8)	Array of characters containing the title of the test	

COMMON/PLOT/

PURPOSE: PLOT stores the constants and characters for the plotting routine

NAME	TYPE	LENGTH	DEFINITION	SET VALUES
NFIRST	I		Position in the arrays of the first ordered pair to be plotted	1
NLAST	I		Position in the arrays of the last ordered pair to be plotted	101
NPOINT	I		NPOINT equals "1" each point from NFIRST to NLAST is to be plotted, "2" if every other point is to be plotted, etc.	1
NMAX	R		Value of abscissa at right-most grid line	1.0
XMIN	R		Value of abscissa at left-most grid line	-1.0
NSCLI	L		Logical value (Should be false if PLOTI has not been called and standard grid is desired)	
NCHAR	I		Number of valid characters in label	
NSCALE	I	(4)	Printing scale factor of ordinate	
PCHAR	I	(2)	Plotting characters	

COMMON/PSI/

PURPOSE: PSI stores the parameters for the integration procedure

NAME	TYPE	LENGTH	DEFINITION	SET VALUES
NPTSZ	I		Number of integration steps from γ_{os} to $\gamma_{os}+1$ (odd number)	31
PTSAF	R		Scaling factor of step size in integrating from α_{max} to $\alpha_{\ell smax}$	10.0
EXPN	R		Empirical constant for intergra-tion to stop	7.0
NALMAX	I		Maximum number of integration steps from $\gamma_{os}+1$ to α_{max}	300
NAL	I		Counter of integration steps	
TAIL	R		Integration made from α to $\alpha\ell s_{max}$	
ALFA	R		Integrating variable (α)	
ALMAX	R		Upper limit of α for integration	
NSTEPS	I		Number of integration steps from α_{max} to $\alpha_{\ell smax}$	

COMMON/XRPLOTF/

PURPOSE: XRPLOTF stores the values of variables for the plotting routine

NAME	TYPE	LENGTH	DEFINITION
XL	R		Value of abscissa at left-most grid line
XH	R		Value of abscissa at right-most grid line
YL	R		Value of ordinate at bottom grid line
YH	R		Value of ordinate at top grid line
XMOV	R		Abscissa index increment numberofor array GRAF
YMOV	R		Ordinate index increment number for array GRAF

COMMON/XRPLOTG/

PURPOSE: XEPLOTG stores the values and characters of variables for the plotting routein

NAME	TYPE	LENGTH	DEFINITION
GRAF	I	(11,204)	Array containing the image to be plotted

COMMON/XRPLOTQ/

PURPOSE: XRPLOTQ stores the constants and characters for the plotting routine

NAME	TYPE	LENGTH	DEFINITION	SET VALUES
II	I		Ordinate scale factor	
JJ	I		Ordinate scale factor	
KK	I		Ordinate scale factor	
LL	I		Ordinate scale factor	
NHL	I		Number of horizontal grid lines	6
NSBH	I		Number of spaces between adjacent horizontal grid lines	10
NVL	I		Number of vertical grid lines	11
NSBV	I		Number of spaces between adjacent vertical grid lines	10
NCHAR	I		Plotting character of horizontal grid lines	1H.
VCHAR	I		Plotting character of vertical grid lines	1H.
IY	I		Number of vertical spaces	
IX	I		Number of horizontal spaces	
V	L		Logical variable, = . TRUE . when the maximum and minimum values of the oridinate are determined	
H	L		Logical variable, = . TRUE . when the maximum and minimum values of the abscissa are determined	

SUBROUTINE DOCUMENTATION FOR CLOSEFIT

NAME:	PROGRAM SWATH
PURPOSE:	Program SWATH is the main program for the CLOSEFIT resistance prediction of Small-Water-plane-Area Twin-Hull Ships
CALLING SERVICE:	Program SWATH (INPUT, OUTPUT, TAPE5 = INPUT TAPE6 = OUTPUT, TAPE8)
ARGUMENTS:	NONE
COMMON BLOCKS:	OUT, AUX, OMEGA, PSI, COEFS
SUBROUTINES CALLED:	CHEB, ECHO, PCHEB, PRINT, RWAVE, RWOUT
COMMENTS:	

NAME: SUBROUTINE BESSJ

PURPOSE: SUBROUTINE BESSJ evaluates the Bessel function from order 0 to order N.

CALLING SEQUENCE: CALL BESSJ (X, N, VJ)

ARGUMENTS:

- X Argument of the Bessel function
- N Maximum order of the Bessel function
- VJ Array holding (N+1) values of the Bessel function of order zero up to N, where

$$VJ(0) = J_0(X)$$

$$\vdots$$

$$VJ(N) = J_N(X)$$

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: RINTEG, RWAVE

COMMENTS:

NAME:	FUNCTION CFITTC
PURPOSE:	Function CFITTC determines the ITTC frictional resistance coefficient C_F
CALLING SEQUENCE:	CF = CFITTC (RN)
ARGUMENT:	RN Reynolds number R_n at test condition
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	RWOUT
COMMENTS:	$C_f = \dfrac{0.075}{[\log_{10}(R_n) - 2]^2}$

NAME:	SUBROUTINE CHEB		
PURPOSE:	Subroutine CHEB computes from offset tables the coefficients of the Chebychev Series, which is an approximation of the outline of the waterplane area or body cross-sectional area curve.		
CALLING SEQUENCE:	CALL CHEB (AMC, BMC, MMAX)		
ARGUMENTS:	AMC	Coefficients of the Chebychev Cosine Series	
	BMC	Coefficients of the Chebychev Sine Series	
	MMAX	Maximum order of the Chebychev Series	
COMMON BLOCKS:	NAVME, OFFSET		
SUBROUTINES CALLED:	READ, SCALE, COMPUT, CHECK, SURFACE		
CALLED BY:	SWATH		
COMMENTS:			

NAME: SUBROUTINE CHEB

PURPOSE: Subroutine CHEB computes from offset tables the coefficients of the Chebychev Series, which is an approximation of the outline of the waterplane area or body cross-sectional area curve.

CALLING SEQUENCE: CALL CHEB (AMC, BMC, MMAX)

ARGUMENTS:

AMC	Coefficients of the Chebychev Cosine Series
BMC	Coefficients of the Chebychev Sine Series
MMAX	Maximum order of the Chebychev Series

COMMON BLOCKS: NAME, OFFSET

SUBROUTINES CALLED: READ, SCALE, COMPUT, CHECK, SURFACE

CALLED BY: SWATH

COMMENTS:

NAME:	SUBROUTINE CHECK	
PURPOSE:	Subroutine CHECK evaluates the Chebychev Series from the Chebychev coefficients obtained from Subroutine COMPUT and prints out the evaluated offset table for manual checking.	
CALLING SEQUENCE:	CALL CHECK (AMC, BMC, MMAX, ISBODY)	
ARGUMENTS:	AMC	Coefficients of the Chebychev Cosine Series
	BMC	Coefficients of the Chebychev Sine Series
	MMAX	Maximum order of the Chebychev polynomial
	ISBODY	Logical flag, indicating "Is this for body?"

$$\text{ISBODY} = \begin{cases} \text{.TRUE. for body} \\ \text{.FALSE. for strut} \end{cases}$$

COMMON BLOCKS:	OFFSET, NAME
SUBROUTINE CALLED:	EVAL
CALLED BY:	CHEB
COMMENTS:	

NAME:	SUBROUTINE CHEV
PURPOSE:	Subroutine CHEV uses the offsets to compute the Chebychev coefficients via spline approximation and numerical integration.
CALLING SEQUENCE:	AMC — Coefficients of the Chebychev Cosine Series
	BMC — Coefficients of the Chebychev Sine Series
	MMAX — Maximum order of the Chebychev polynomial
	ISBODY — Logical flag, indicating "Is this for body?"
	ISBODY = { . TRUE . for body; . FALSE . for strut }
COMMON BLOCKS:	OFFSET, NAME
SUBROUTINES CALLED:	SPLINE, FUNCTION SUMPSPL
CALLED BY:	COMPUT
COMMENTS:	

NAME:	SUBROUTINE COMPUT	
PURPOSE:	Subroutine COMPUT initiates the computation of the Chebychev coefficients.	
CALLING SEQUENCE:	CALL COMPUT (AMC, BMC, MMAX, ISBODY)	
ARGUMENTS:	AMC	Coefficients of the Chebychev Cosine Series
	BMC	Coefficients of the Chebychev Sine Series
	MMAX	Maximum order of the Chebychev polynomials
	ISBODY	Logical flag, indicating "Is this for body?"

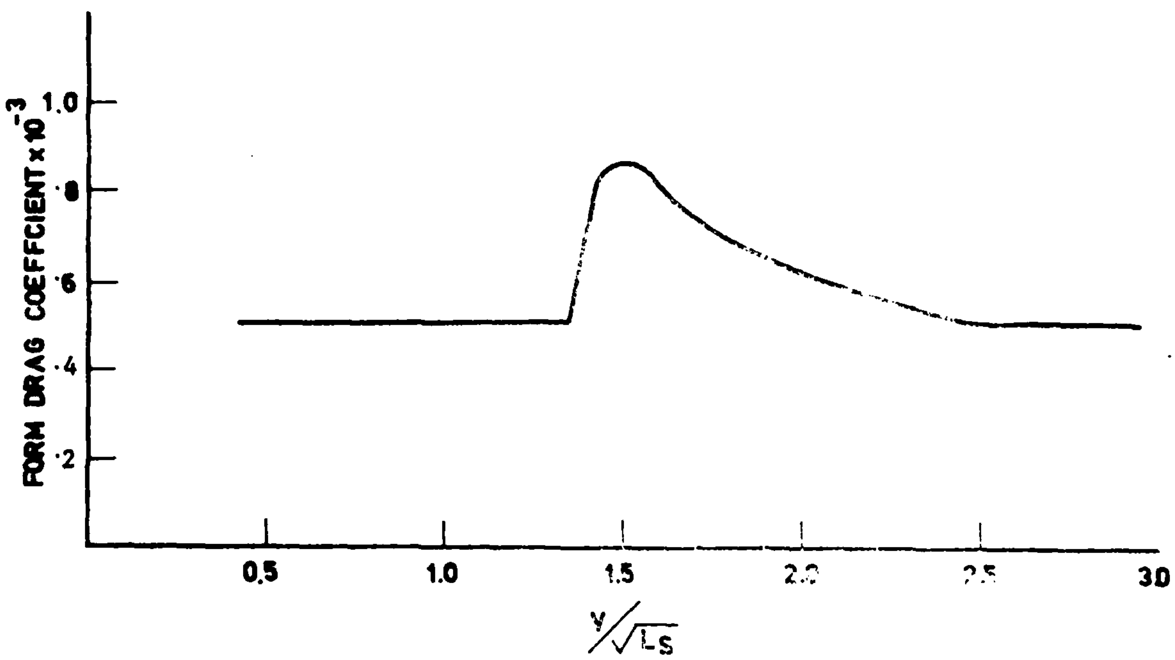
$$\text{ISBODY} = \begin{cases} .\text{TRUE}. & \text{for body} \\ .\text{FALSE}. & \text{for strut} \end{cases}$$

COMMON BLOCKS:	OFFSET, NAME, NORM
SUBROUTINE CALLED:	CHEV
CALLED BY:	CHEB
COMMENTS:	

NAME:	SUBROUTINE ECHO
PURPOSE:	Subroutine ECHO prints a summary of input data and the Chebychev coefficients
CALLING SEQUENCE:	CALL ECHO (NLOC)
ARGUMENT:	NLOC Number of strut locations to be tested
COMMON BLOCKS:	AUX, OMEGA, OUT
SUBROUTINE CALLED:	NONE
CALLED BY:	SWATH
COMMENTS:	

NAME:	SUBROUTINE EVAL		
PURPOSE:	Subroutine EVAL evaluates the Chebychev Series at a given station.		
CALLING SEQUENCE:	CALL EVAL (XX, MMAX, FFSET, AMC, BMC)		
ARGUMENTS:	XX	Value of X station on a range of -1 to +1	
	MMAX	Maximum order of the Chebychev polynomial	
	FFSET	Offset approximated by the Chebychev Series	
	AMC	Coefficients of the Chebychev Cosine Series	
	BMC	Coefficients of the Chebychev Sine Series	
COMMON BLOCKS:	NONE		
SUBROUTINE CALLED:	NONE		
CALLED BY:	CHECK		
COMMENTS:			

NAME:	FUNCTION FORMDR	
PURPOSE:	Function FORMDR evaluates the form drag coefficient, based on experimental data	
CALLING SEQUENCE:	FOR = FORMDR (VL)	
ARGUMENT:	VL	Ratio of ship speed to length of strut $V/\sqrt{L_S}$
COMMON BLOCKS:	NONE	
SUBROUTINE CALLED:	FUNCTION YINTP	
CALLED BY:	ROUT	
COMMENTS:	Form drag is assumed to be 0.5×10^{-3} for $V/\sqrt{L_S}$ less than 1.325 and greater than 2.5	



NAME:	SUBROUTINE PCHEB
PURPOSE:	Subroutine PCHEB plots by line printer the body sectional area curve and waterplane outline curve from the given Chebychev coefficients
CALLING SEQUENCE:	CALL PCHEB (AS, BS, AB, BB, NN, TITLE)
ARGUMENTS:	AS Coefficients of Chebychev Sine Series for strut
	BS Coefficients of Chebychev Cosine Series for strut
	AB Coefficients of Chebychev Sine Series for body
	BB Coefficient of Chebychev Cosine Series for body
	NN Maximum order of Chebychev Series
	TITLE Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCKS:	PLOT, XRPLOTQ
SUBROUTINES CALLED:	PLOT1, PLOT2, PLOT3, PLOT4
CALLED BY:	CHEB
COMMENTS:	

NAME:	SUBROUTINE PLOT1		
PURPOSE:	Subroutine PLOT1 sets up spacing and determines the values of the axes.		
CALLING SEQUENCE:	CALL PLOT 1 (NSCALE, A, B, C, D, E, F)		
ARGUMENTS:	NSCALE		Integer array defined as follows:
	NSCALE (1)	=	I, if printed values of the ordinate are 10 ** I times the actual value
	NSCALE (2)	=	J, if printed values of the ordinate are 10 ** J times the actual value
	NSCALE (3)	=	K, if printed values of the abscissa are 10 ** K times the actual values
	NSCALE (4)	=	L, if printed values of the abscissa are 10 ** L times the actual value
	A		Integer number of horizontal grid lines
	B		Integer number of spaces beyond each horizontal grid line to the next grid line
	C		Integer number of vertical grid lines
	D		Integer number of spaces beyond each vertical grid line to the next grid line
	E		Horizontal grid character
	F		Vertical grid character
COMMON BLOCK:	XRPLOTQ		
SUBROUTINE CALLED:	NONE		
CALLED BY	PCHEB		
COMMENTS:			

NAME:	SUBROUTINE PLOT2
PURPOSE:	Subroutine PLOT2 examines the minimum and maximum values of the abscissa and the ordinate and establishes an internal formula for computing location in the image region corresponding to the point to be plotted.
CALLING SEQUENCE:	CALL PLOT2 (XMAX, XMIN, YMAX, YMIN, NSCLI)
ARGUMENTS:	XMAX — Value of abscissa at right-most grid line
	XMIN — Value of abscissa at left-most grid line
	YMAX — Value of ordinate at top grid line
	YMIN — Value of ordinate at bottom grid line
	NSCLI — Logical flat (should be .FALSE., if PLOT1 has not been called and standard grid is desired)
COMMON BLOCKS:	XRPLOTF, SRPLOTQ, XRPLOTG
SUBROUTINE CALLED:	NONE
CALLED BY:	PCHEB
COMMENTS:	

NAME:	SUBROUTINE PLOT3	
PURPOSE:	Subroutine PLOT3 assigns an alpha-character to each point to be plotted.	
CALLING SEQUENCE:	CALL PLOT3 (PCHAR, X, Y, SDATA, FDATA, DDATA)	
ARGUMENTS:	PCHAR	Plotting character
	X	Array containing the X coordinates to be plotted
	Y	Array containing the Y coordinates to be plotted
	SDATA	Integer position in the arrays of the first ordered pair to be plotted
	FDATA =	1 if each point from SDATA to DDATA is to be plotted 2 if every other point is to be plotted 3 if every third point is to be plotted
	DDATA	Integer position in the array of the last ordered pair to be plotted
COMMON BLOCKS:	XRPLOTF, XRPLOTG	
SUBROUTINE CALLED:	*NONE*	
CALLED BY:	PCHEB	
COMMENTS:		

NAME:	SUBROUTINE PLOT4
PURPOSE:	Subroutine PLOT4 prints the image of the completed graph on the printer, including the values of the abscissa and the ordinate at the grid lines outside the bottom and left edge of the graph.
CALLING SEQUENCE:	CALL PLOT4 (MCHAR, NCHAR)
ARGUMENTS:	MCHAR Single dimension array containing alpha-characters to be plotted at the left of the graph NCHAR Number of valid characters in MCHAR
COMMON BLOCKS:	XRPLOTF, XRPLOTG, XRPLOTQ
SUBROUTINE CALLED:	QPLOTZ5
CALLED BY:	PCHEB
COMMENTS:	

NAME:	SUBROUTINE PRINT
PURPOSE:	Subroutine PRINT prints the auxiliary function tables of T and W.
CALLING SEQUENCE:	CALL PRINT (I, NLOC2)
ARGUMENTS:	I Index indicating strut location NLOC2 Flag indicating the presence of a second strut NLOC2 = {0 if single strut; 1 if tandem struts}
COMMON BLOCKS:	OUT, AUX, OMEGA
SUBROUTINE CALLED:	NONE
CALLED BY:	SWATH
COMMENTS:	

NAME:	SUBROUTINE QPLOTZ5
PURPOSE:	Subroutine QPLOTZ5 calculates the scaling information needed to generate the format to label the left-hand side of the program.
CALLING SEQUENCE:	CALL QPLOTZ5 (PDQ)
ARGUMENT:	PDQ Scaling factor for ordinate plot
COMMON BLOCKS:	XRPLOTF, XRPLOTQ
SUBROUTINES CALLED:	NONE
COMMENTS:	

NAME:	SUBROUTINE READ
PURPOSE:	Subroutine READ reads the offset data of the strut or body and checks for error.
CALLING SEQUENCE:	CALL READ
ARGUMENTS:	NONE
COMMON BLOCKS:	OFFSET, NAME, NORM
SUBROUTINE CALLED:	NONE
CALLED BY:	CHEB
COMMENTS:	

NAME:	SUBROUTINE REFLKT
PURPOSE:	Subroutine REFLKT reflects all the symmetrical matrices of T and W.
CALLING SEQUENCE:	CALL REFLKT
ARGUMENTS:	NONE
COMMON BLOCKS:	AUX, OMEGA
SUBROUTINE CALLED:	NONE
CALLED BY:	RWAVE
COMMENTS:	

NAME:	SUBROUTINE RINIT
PURPOSE:	Subroutine RINIT initializes the T and W arrays to zero.
CALLING SEQUENCE:	CALL RINIT
ARGUMENT:	NONE
COMMON BLOCKS:	AUX, OMEGA
SUBROUTINE CALLED:	NONE
CALLED BY:	RWAVE
COMMENTS:	

NAME: SUBROUTINE RINTEG

PURPOSE: Subroutine RINTEG evaluates the integrand for T and W functions.

CALLING SEQUENCE: CALL RINTEG (ALFA, B, D, NLOC2, WTINT, SEPCOS, SQ)

ARGUMENTS:

ALFA	Integrating variable α
B	CSTRUT(I)/XLS
D	CSTRT2(I)/XLS
NLOC2	Flag indicating the presence of a second strut
	$NLOC2 = \begin{cases} 0 \text{ if single strut} \\ 1 \text{ if tandem struts} \end{cases}$
WTINT	Weighting constant for the integrand
SEPCOS	Value of the cosine function in the integrand
SQ	Value of part of the integrand

$$\frac{1}{(\alpha^2 - \gamma^2)^{\frac{1}{2}}}$$

COMMON BLOCKS: AUX, OMEGA

SUBROUTINE CALLED: BESSJ

CALLED BY: RWAVE

COMMENTS:

NAME: SUBROUTINE RWAVE

PURPOSE: Subroutine RWAVE computes the auxiliary function of T and W.

CALLING SEQUENCE: CALL RWAVE (B, D, NLOC2)

ARGUMENTS:

B CSTRUT (I)/XLS
D CSTRT2 (I)/XLS
NLOC2 Flag indicating the presence of a second strut

$$\text{NLOC2} = \begin{cases} 0 & \text{if single strut} \\ 1 & \text{if tandem struts} \end{cases}$$

COMMON BLOCKS: OUT, AUX, OMEGA, PSI

SUBROUTINES CALLED: RINIT, SIMPSN, RINTEG, BESSJ, REFLKT

CALLED BY: SWATH

COMMENTS:

Typical Auxiliary Functions

$$\left\{ \begin{array}{c} \dfrac{T_{Smn}}{(2m-1)(2n-1)} \\ \dfrac{W_{Smn}}{(2m)(2n)} \end{array} \right\} = \int_{\gamma_{0S}}^{\infty} \frac{d\alpha}{\alpha^2 \sqrt{\alpha^2 - \gamma_{0S}^2}} \, D\left(\alpha, \frac{2b}{L_S}, \gamma_{0S}\right) \times E_S^2(\alpha) \left\{ \begin{array}{l} J_{2m-1}(\alpha) J_{2n-1}(\alpha) \\ J_{2m}(\alpha) J_{2n}(\alpha) \end{array} \right\},$$

where

$$D = 1 + \cos\left[\left(\frac{2b}{L_S}\right)\left(\frac{2}{\gamma_{0S}}\right)\alpha\sqrt{\alpha^2 - \gamma_{0S}^2}\right],$$

$$E_S = 1 - e^{-2(h_S/L_S)(\alpha^2/\gamma_{0S})},$$

$$\gamma_{0S} = \frac{gL_S}{(2U^2)} .$$

NAME:	SUBROUTINE RWOUT
PURPOSE:	Subroutine RWOUT computes and prints drag coefficients and total resistance of the test model.
CALLING SEQUENCE:	CALL RWOUT (I, NLOC2)
ARGUMENTS:	I — Index indicating strut location NLOC2 — Flag indicating the presence of a second strut $NLOC\ 2 = \begin{cases} 0 \text{ if single strut} \\ 1 \text{ if tandem struts} \end{cases}$
COMMON BLOCKS:	OUT, AUX, OMEGA
SUBROUTINES CALLED:	SUM, FUNCTION FORMDR, FUNCTION CFITTC
CALLED BY:	SWATH
COMMENTS:	

NAME: SUBROUTINE SCALE

PURPOSE: Subroutine SCALE computes scaling factors and scales the input for length between -1.0 and +1.0 maximum beam and sectional area to 1.0.

CALLING SEQUENCE: CALL SCALE (ISBODY)

ARGUMENT: ISBODY Logical flag indicating, "Is this for body?"

$$\text{ISBODY} = \begin{cases} .\ \textbf{TRUE}\ .\ \text{for body} \\ .\ \textbf{FALSE}\ .\ \text{for strut} \end{cases}$$

COMMON BLOCKS: OFFSET, NAME, NORM

SUBROUTINE CALLED: NONE

CALLED BY: CHEB

COMMENTS:

NAME:	SUBROUTINE SIMPSN
PURPOSE:	Subroutine SIMPSN sets up Simpson's multipliers for numerical integration by Simpson's rule.
CALLING SEQUENCE:	CALL SIMPSN (NPTS, SIMP)
ARGUMENTS:	NPTS Number of integration steps SIMP Array containing Simpson's multipliers
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	RWAVE
COMMENTS:	

NAME: SUBROUTINE SPLINE

PURPOSE: Subroutine SPLINE fits smooth spline segments through a given set of discrete data points.

CALLING SEQUENCE: CALL SPLINE (X, Y, D, N, BC, KODE, KKODE, ISBODY)

ARGUMENTS:

X	Value of abscissas of offset data
Y	Value of ordinates of offset data
D	Value of second derivatives at a point
N	Total number of offset data points
BC	Value of boundary conditions
KODE =	0 if user is specifying boundary conditions 1 if user is taking an extrapolation of second derivatives as boundary conditions
KKODE =	0 if there is no printout from spline routine 1 if spline prints abscissas, ordinates and second derivatives
ISBODY	Logical flag indicating, "Is this for body?"
ISBODY =	. TRUE . for body . FALSE . for strut

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: CHEV

COMMENTS:

NAME:	SUBROUTINE SUM
PURPOSE:	Subroutine SUM computes sums of the form:

$$\sum_{m=1}^{M} \sum_{n=1}^{N} \{A_{sm} A_{sn} T_{smn} + B_{sm} B_{sn} W_{smn}\}$$

CALLING SEQUENCE: CALL SUM (L, SUM1S, SUM1B, SUM1SB, SUM12S, SUM12B, SUM12SB)

ARGUMENTS:

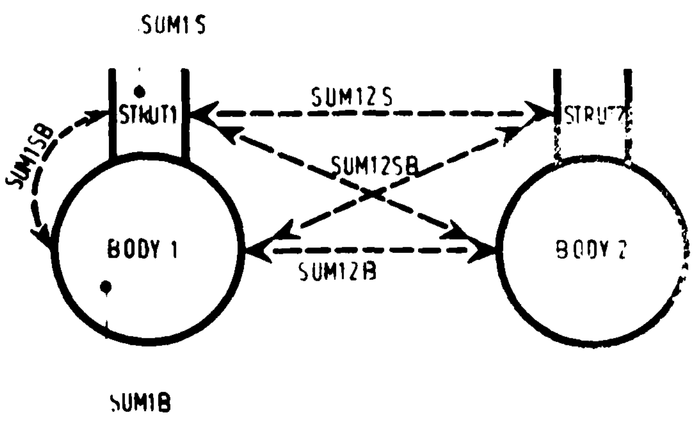
L	Number of different hull separation distances
SUM1S	Partial sum for strut 1
SUM1B	Partial sum for body 1
SUM1SB	Partial sum for interaction between strut 1 and body 1
SUM12S	Partial sum for interaction between strut 1 and strut 2
SUM12B	Partial sum for interaction between body 1 and body 2
SUM12SB	Partial sum for interactions between strut 1 and body 2 or strut 2 and body 1

COMMON BLOCKS: AUX, OMEGA, COEFS

SUBROUTINE CALLED: NONE

CALLED BY: RWOUT

COMMENTS:



NAME:	FUNCTION SUMSPL
PURPOSE:	Function SUMSPL uses the spline coefficients and integrates the trigonometric integrals F(X)*SIN(AK*X) or F(X)*COS(AK*X).
CALLING SEQUENCE:	SUM = SUMSPL (X, Y, S, N, AK, KODE)
ARGUMENTS:	X Values of abscissas of offset data Y Values of ordinates of offset data S Values of second derivatives N Total number of data points AK (2m-1) or (2m), m = 1,2,3 . . . MMAX KODE = { +1 if sine integration desired; -1 if cosine integration desired
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	CHEV

NAME:	SUBROUTINE SURFACE
PURPOSE:	Subroutine SURFACE finds the wetted surface area of the strut or body as well as the waterplane area or displaced volume.
CALLING SEQUENCE:	CALL SURFACE (AMC, BMC, MMAX, ISBODY)
ARGUMENTS:	AMC — Coefficients of the Chebychev Cosine Series
	BMC — Coefficients of the Chebychev Sine Series
	MMAX — Maximum order of the Chebychev polynomials
	ISBODY — Logical flag indicating "Is this for body?"
	$\text{ISBODY} = \begin{cases} \text{.TRUE. for body} \\ \text{.FALSE. for strut} \end{cases}$
COMMON BLOCKS:	OUT, NAME
SUBROUTINES CALLED:	WTSURFS, WTSURFB
CALLED BY:	CHEB
COMMENTS:	

NAME:	SUBROUTINE WTSURFB
PURPOSE:	Subroutine WTSURFB determines the surface area of a body of revolution given by the Chebychev coefficients.
CALLING SEQUENCE:	CALL WTSURFB (AMC, BMC, MMAX)
ARGUMENTS:	AMC — Coefficients of the Chebychev Cosine Series
	BMC — Coefficients of the Chebychev Sine Series
	MMAX — Maximum order of the Chebychev Series
COMMON BLOCK:	OUT
SUBROUTINE CALLED:	EVAL
CALLED BY:	SURFACE
COMMENTS:	

NAME:	SUBROUTINE WTSURFS
PURPOSE:	Subroutine WTSURFS determines the wetted surface area of a strut whose thickness distribution is given by the Chebychev coefficients.
CALLING SEQUENCE:	CALL WTSURFS (AMC, BMC, MMAX)
ARGUMENTS:	AMC — Coefficients of Chebychev Cosine Series BMC — Coefficients of Chebychev Sine Series MMAX — Maximum order of Chebychev Series
COMMON BLOCK:	OUT
SUBROUTINE CALLED:	EVAL
CALLED BY:	SURFACE
COMMENTS:	

NAME:	FUNCTION YINTP
PURPOSE:	Function YINTP interpolates through a set of discrete data.
CALLING SEQUENCE:	FR = YINTP (XA, X, Y, N)
ARGUMENTS:	XA Point to be interpolated X Array of ordinate data Y Array of abscissa data N Number of data points
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	FORMDR
COMMENTS:	Uses linear interpolation

PROGRAM DOCUMENTATION OF SYNTHESIS

COMMON BLOCK DEFINITIONS FOR SYNTHESIS

COMMON/AUX/

PURPOSE: AUX stores the auxiliary wave resistance function table.

NAME	TYPE	LENGTH	DEFINITION
TS	R	(3,3)	Auxiliary function T for strut
WS	R	(3,3)	Auxiliary function W for strut
TB	R	(3,3)	Auxiliary function T for body
WB	R	(3,3)	Auxiliary function W for body
TSB	R	(3,3)	Auxiliary function T for strut to body
WSB	R	(3,3)	Auxiliary function W for strut to body
TS12	R	(3,3)	Auxiliary function T for strut 1 to strut 2
WS12	R	(3,3)	Auxiliary function W for strut 1 to strut 2
TB12	R	(3,3)	Auxiliary function T for body 1 to body 2
WB12	R	(3,3)	Auxiliary function W for body 1 to body 2
TSB12	R	(3,3)	Auxiliary function T for strut 1 to body 2
WSB12	R	(3,3)	Auxiliary function W for strut 1 to body 2
TSBP	R	(3,3)	Auxiliary function T
WSBP	R	(3,3)	Auxiliary function W
TSB12P	R	(3,3)	Auxiliary function T
WSB12P	R	(3,3)	Auxiliary function W

COMMON/BARCK/IE /

PURPOSE: BARCK stores the variables which define the propulsor design condition.

NAME	TYPE	LENGTH	DEFINITION
IE	I		Error message pointer

$$IE = \begin{cases} 0, \text{if program ran in normal fashion} \\ 100, \text{if limiting value of P/D has been used} \\ 200, \text{if there is an iteration malfunction -- PD(2) used for P/D} \\ 3, \text{if maximum BAR, 0.95, used -- marine screw} \\ 4, \text{if minimum BAR, .4, used -- marine screw} \\ 5, \text{if maximum tip speed used -- air screw} \\ 6, \text{if blade number is outside program limit -- closest number used} \end{cases}$$

COMMON/CDESC/

PURPOSE: CDESC stores the values of constants which define water properties.

NAME	TYPE	LENGTH	DEFINITION
RHOD	R	(3)	Water density array
ANUD	R	(3)	Kinematic viscosity array
IP	I		Pointer indicating the particular water condition used. Set IP = 1 for this program to obtain sea water condition.

COMMON/COEFS/

PURPOSE: COEFS stores the values of the Chebyehev coefficients for the strut and body.

NAME	TYPE	LENGTH	DEFINITION
ASM	R	(3)	Coefficients of Chebychev Sine Series for strut
BSM	R	(3)	Coefficients of Chebychev Cosine Series for strut
ABM	R	(3)	Coefficients of Chebychev Sine Series for body
BBM	R	(3)	Coefficients of Chebychev Cosine Series for body
MMAX	I		Maximum order of Chebychev Series

COMMON/CPROP/

PURPOSE: CPROP stores the values of the variables and constants used in the propulsor design routine.

NAME	TYPE	LENGTH	DEFINITION
IPTYP	I	(3)	Indicating propeller type
DIAM	R	(3)	Diameter of propeller (ft)
VADES	R	(3)	Speed of advance of propulsor at design condition (kts)
RPMDES	R	(3)	RPM of propeller at design point
TDES	R	(3)	Thrust of propulsor system (lbs)
QDES	R	(3)	Torque of propulsor system (ft-lbs)
EFFDES	R	(3)	Efficiency of propulsor at design condition
OMWTD	R	(3)	Thrust wake $(1-w_T)$
CAV	R	(3)	Percent back cavitation allowed on water screw
Z	I	(3)	Number of blades of propeller
H	R	(3)	Depth of water to centerline of propulsor shaft
BAR	R	(3)	Blade area ratio
PD	R	(3)	P/D of propulsor
J	R	(3)	J of propulsor
IERROR	I	(3)	Error message indicator
KT	R	(3)	Thrust coefficient (K_T)
KQ	R	(3)	Torque coefficient (K_Q)

COMMON/INPUT/

PURPOSE: INPUT stores input data.

NAME	TYPE	LENGTH	DEFINITION
XLS	R		Length of strut L_S (ft)
HS	R		Draft of strut H_S (ft)
TSMAX	R		Maximum thickness of strut t_{max} (ft)
CWP	R		Waterplane area coefficient $C_{WP} = A_W/(L_S\ t_{max})$ where A_W = waterplane are of one strut
CLCF	R		Waterplane moment coefficient $C_{LCF} = Mx_s/(A_w \cdot L_s)$*
CIYY	R		Waterplane inertia coefficient $C_{IYY} = I_{wp}/(A_w \cdot L_s^2)$*
XLB	R		Length of body L_S (ft)
HB	R		Draft of body H_S (ft)
AX	R		Maximum cross-sectional area of body A_X (ft^2)
CP	R		Body prismatic coefficient $C_P = \nabla_b/(A_X\ L_b)$ where ∇_b = displaced volume of one body
CLCB	R		Body moment coefficient $C_{LCB} = Mx_B/(\nabla_e \cdot L_e)$*
SEPDIS	R		Separation distance between centerline of the two bodies (ft)
CSTRUT	R		Distance from centerline of strut to centerline of body (ft)
CSTRT2	R		Distance from centerline of second strut to centerline of body (ft)
PDIA	R		Diameter of propeller D (ft)
SPAN	R		Span of a demi-fin (ft)
CHORD	R		Chord of a fin (ft)
TFINS	R		Maximum thickness of a demi-fin (ft)
NLOC	I		Switch indicating the presence of a second strut

*All moments and moments of inertia are taken about the mid-length of the respective strut or body.

COMMON/OFFDES/

PURPOSE: OFFDES stores the variables which define the coefficients of KT/KQ curves.

NAME	TYPE	LENGTH	DEFINITION
PJAY	R	(4)	Coefficients of polynomial representing KT or KQ.

COMMON/OMEGA/

PURPOSE: OMEGA stores the values of variables and constants for evaluation of auxiliary functions and ship resistance coefficients.

NAME	TYPE	LENGTH	DEFINITION
VMFPS	R		Speed of ship V_K (kts)
GAMAOS	R		$\gamma_{os} = \frac{1}{2}\, g\, L_S/V^2$ where U is speed in fps
GAMAOB	R		$\gamma_{ob} = \frac{1}{2}\, g\, L_b/V^2$
GOSQ	R		$(\gamma_{os})^2$
HSOLS	R		Ratio of draft to length of strut H_S/L_S
HBOLB	R		Ratio of draft to length of body H_b/L_b
WETS	R		Wetted surface area of strut S_s (ft^2)
WETB	R		Wetted surface area of body S_b (ft^2)
WETFIN	R		Wetted surface area of a fin S_f (ft^2)
WTSURF	R		Total wetted surface area
SEP	R		$2b/(\gamma_{os} \cdot L_s)$
PHIS	R		$2(\frac{h_s}{L_b})/\gamma_{os}$
PHIB	R		$2(\frac{h_s}{L_s})/\gamma_{ob}$
RATIOL	R		γ_{ob}/γ_{os}
CFS	R		Frictional drag coefficient of strut C_{F_S}
CFB	R		Frictional drag coefficient of body C_{F_b}
APPDRG	R		Appendage drag (lb)

COMMON/PCOF /

PURPOSE: PCOF stores coefficients for KQ/KT polynomial approximation.

NAME	TYPE	LENGTH	DEFINITION
CT	R	(4)	Coefficients of the polynomial which is an approximation of the KT curve
CQ	R	(4)	Coefficients of the polynomial which is an approximation of the KQ curve

COMMON/PHYSCO/

PURPOSE: PHYSCO stores physical constants.

NAME	TYPE	LENGTH	DEFINITION
RHO	R		Density of water
GNU	R		Kinematic viscosity of water
G	R		Acceleration due to gravity
PI	R		Ratio of circumference to diameter of a circle
DELCF	R		Correction allowance of friction drag from ITTC line.

COMMON/PLOT/

PURPOSE: PLOT stores the constants and characters for the plotting routine.

NAME	TYPE	LENGTH	DEFINITION
NFIRST	I		Position in the arrays of the first ordered pair to be plotted
NLAST	I		Position in the arrays of the last ordered pair to be plotted
NPOINT	I		NPOINT equals "1" if each point from NFIRST to NLAST is to be plotted, "2" if every other point is to be plotted, etc.
XMAX	R		Value of abscissa at right-most grid line
XMIN	R		Value of abscissa at left-most grid line
NSCLI	L		Logical value (Should be false if PLOT1 has not been called and standard grid is desired)
NCHAR	I		Number of valid characters in label
NSCALE	I	(4)	Printing scale factor of ordinate
PCHAR	I	(2)	Plotting characters

COMMON/PSI/

PURPOSE: PSI stores the values of the variables and constants used in integration routines.

NAME	TYPE	LENGTH	DEFINITION
NPTSZ	I		Number of integration steps from γ to $\gamma+1$
PTSAF	R		Scaling factor of step size in integrating from α_{max} to $\alpha_{\ell smax}$
EXPN	R		Empirical constant for integration to stop
NALMAX	R		Maximum number of integration steps from $\gamma+1$ to α_{max}
NAL	I		Counter of integration steps
TAIL	R		Integration correction made from α_{max} to $\alpha_{\ell smax}$
ALFA	R		Integrating variable (α)
ALSMAX	R		Maximum of α for integration
NSTEPS	I		Number of integration steps from α_{max} to $\alpha_{\ell smax}$

COMMON/ROOTC/

PURPOSE: ROOTC stores the coefficients of a third order polynomial which is the approximation of the KT curve of a propeller.

NAME	TYPE	LENGTH	DEFINITION
A	R	4	Array holding the coefficient of a third order polynomial which is the approximation of the KT curve of a propeller

COMMON/XRPLOTF/

PURPOSE: XRPLOTF stores the values of variables for the plotting routine.

NAME	TYPE	LENGTH	DEFINITION
XL	R		Value of abscissa at left-most grid line
XH	R		Value of abscissa at right-most grid line
YL	R		Value of ordinate at bottom grid line
YH	R		Value of ordinate at top grid line
XMOV	R		Abscissa index increment number for array GRAF
YMOV	R		Ordinate index increment number for array GRAF

COMMON/XRPLOTG/

PURPOSE: XRPLOTG stores the values and characters of variables for the plotting routine.

NAME	TYPE	LENGTH	DEFINITION
GRAF	I	(11,204)	Array containing the image to be plotted

COMMON/XRPLOTQ

PURPOSE: XRPLOTQ stores the constants and characters for the plotting routine.

NAME	TYPE	LENGTH	DEFINITION
I	I		Ordinate scale factor is 10^I
J	I		Number digits following ordinate decimal point
K	I		Abscissa scale factor is 10^K
L	I		Number digits following abscissa decimal point
NHL	I		Number of horizontal grid lines
NSBH	I		Number of spaces between adjacent horizontal grid lines
NVL	I		Number of vertical grid lines
NSBV	I		Number of spaces between adjacent vertical grid lines
HCHAR	I		Plotting character of horizontal grid lines
ISX	I		Number of horizontal spaces
ISY	I		Number of vertical spaces
V	L		Logical variable, = . TRUE . when the maximum and minimum values of the ordinate are determined
H	L		Logical variable, = . TRUE . when the maximum and minimum values of the abscissa are determined

SUBROUTINE DOCUMENTATION FOR SYNTHESIS

NAME: PROGRAM SYNTH

PURPOSE: Program SYNTH is the main program for calculating the propulsor resistance and design for SWATH ships.

CALLING SEQUENCE: PROGRAM SYNTH (INPUT, OUTPUT, TAPE 5 = INPUT, TAPE 6 = OUTPUT, TAPE 8)

ARGUMENTS: NONE

COMMON BLOCKS: COEFS, INPUT, OMEGA, PHYSCO, PSI

SUBROUTINES CALLED: RIN, CHEB, RWAVE, FINDRG, ROUT, SHPCMP, Function CFITTC

COMMENTS:

NAME:	SUBROUTINE BESSJ
PURPOSE:	Subroutine BESSJ evaluates the Bessel function from order 0 to order N.
CALLING SEQUENCE:	CALL BESSJ (X, N, VJ)
ARGUMENTS:	X Argument of the Bessel function N Maximum order of the Bessel function VJ Array holding (N+1) values of the Bessel function of order zero up to N, where

$$VJ(0) = J_o(X)$$

$$\vdots$$

$$VJ(N) = J_N(X)$$

COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	RINTEG, RWAVE
COMMENTS:	

NAME:	SUBROUTINE BURIL
PURPOSE:	SUBROUTINE BURIL uses Burrill's cavitation criteria to determine if the blade area ratio (BAR) used is sufficient and, if it is not, to estimate what BAR is necessary.
CALLING SEQUENCE:	CALL BURIL (T, D, H, VA, N, CAVI, BAR, IERROR, PD, NRET)

ARGUMENTS:

T	Thrust of propeller (lbs)
D	Diameter of propeller (ft)
H	Depth of propeller shaft (ft)
VA	Speed of advance (fps)
N	RPM of propeller
CAVI	Percent back cavitation allowed
BAR	Blade area ratio
IERROR	Error message indication as defined in Common Block BARCK
PD	P/D of propeller
NRET	Internal flag indicating BAR optimization

$$\text{NRET} = \begin{cases} 0 \text{ if not optimized} \\ 1 \text{ if optimized} \end{cases}$$

COMMON BLOCK:	BARCK
SUBROUTINE CALLED:	NONE
CALLED BY:	PRODES
COMMENTS:	

NAME:	FUNCTION CALC
PURPOSE:	FUNCTION CALC calculates "KT" or "KQ" for two to seven-bladed propellers, given J, BAR, P/D, Z, and an indicator for "KT" or "KQ."
CALLING SEQUENCE:	TQ = CALC (J, BAR, PD, Z, ITQ)
ARGUMENTS:	J Advance coefficient of propeller BAR Blade area ratio PD P/D of propeller Z Number of blades ITQ Index defined by

$$\text{ITQ} = \begin{cases} 1 \text{ if thrust is input} \\ 2 \text{ if power is input} \end{cases}$$

COMMON BLOCK:	OFFDES
SUBROUTINE CALLED:	NONE
CALLED BY:	TROOST, ROOT, POLCOF
COMMENTS:	

NAME:	FUNCTION CFITTC
PURPOSE:	Function CFITTC determines the ITTC frictional resistance coefficient C_F
CALLING SEQUENCE:	CF = CFITTC (RN)
ARGUMENT:	RN Reynolds number at R_n at test conditions
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	SYNTH
COMMENTS:	$C_F = \dfrac{0.075}{[\log_{10}(R_n)-2]^2}$

NAME:	SUBROUTINE CHEB
PURPOSE:	Subroutine CHEB determines the Chebychev coefficients and wetted surface area of the strut and body. It also plots the strut waterline and the body area distribution.
CALLING SEQUENCE:	CALL CHEB (TITLE)
ARGUMENT:	TITLE Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCKS:	COEFS, INPUT, OMEGA, PHYSCO
SUBROUTINES CALLED:	WSURFB, WSURFS, PCHEB
CALLED BY:	SYNTH
COMMENTS:	

NAME:	SUBROUTINE ERROR
PURPOSE:	Subroutine ERROR prints the error message of the propulsor design.
CALLING SEQUENCE:	CALL ERROR (IE)
ARGUMENT:	IE Error message pointer as defined in Common Block BARCK
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	PROPP
COMMENTS:	

NAME: FUNCTION EVAL

PURPOSE: Function EVAL evaluates the Chebychev Series:

$$F(X) = \sum_{M=1}^{MMAX} A(M) * U(M,X) + B(M) * V(M,X)$$

CALLING SEQUENCE: EV = EVAL (X,A,B,MMAX)

ARGUMENTS:

X	Arguments of the Chebychev Series
A	Coefficients of the Chebychev Cosine Series
B	Coefficients of the Chebychev Sine Series
MMAX	Maximum order of the Chebychev Series

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: WSURFS, WSURFB

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: WSURFS, WSURFB

COMMENTS:

$U_M(X) = \mathrm{Cos}[(2M-1)(\theta)]$

$V_M(X) = \mathrm{Sin}\ 2M\ \theta$

$\theta = \mathrm{Sin}^{-1}(X)$

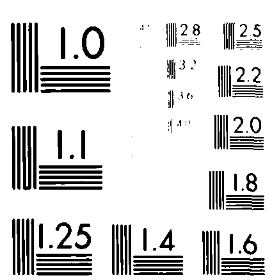
NAME:	SUBROUTINE FINDRG
PURPOSE:	Subroutine FINDRG determines the fin drag.
CALLING SEQUENCE:	CALL FINDRG (RHOS, XNUS, VSFW, SFINS, TFINS, CFINS, TBFINS, SBFINS, DS)
ARGUMENTS:	RHOS Density of water XNUS Kinematic viscosity of water VSFW Ship speed (fps) SFINS Chord * span TFINS Thickness of a fin (ft) CFINS Chord of a fin (ft) TBFINS 0.01 * TFINS SBFINS TBFINS * span DS Total fin drag
COMMON BLOCKS:	NONE
SUBROUTINES CALLED:	FRICT, FOILSC
CALLED BY:	SYNTH
COMMENTS:	

AD-A099 533 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 20/4
DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTI--ETC(U)
APR 81 A M REED
UNCLASSIFIED DTNSRDC/SPD-0927-02 NL

2 OF 3
AD A
099533

2 OF 3

AD A
099533



MICROCOPY RESOLUTION TEST CHART

NAME:	SUBROUTINE FOILSC
PURPOSE:	Subroutine FOILSC calculates the drag components for a symmetrical foil section.
CALLING SEQUENCE:	CALL FOILSC (XS, CFS, TOC, TOC2, TOC4, SS, TS2, SBS, CLOTB, RFS, RFVAS, RPS, RINTS, RBS)

ARGUMENTS:

XS	Ship constant = ½ * ρ * SFINS VSFW ** 2
CFS	Flat plate friction coefficient
TOC	Thickness/chord ratio
TOC2	T/C**2
TOC4	T/C**4
SS	Chord * span ≃ area of foil
TS2	(SS) ** 2 ≃ area squared
SBS	TBFINS * span
CLOTB	Chord/0.01 * TFINS
RFS	Frictional resistance of a flat plate
RFUAS	Resistance due to velocity argumentation
RPS	Resistance due to pressure and separation
RINTS	Resistance due to intersection with hull
RBS	Base drag due to bluntness of trailing edge

COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	FINDRG
COMMENTS:	

NAME: FUNCTION FORMDR

PURPOSE: Function FORMDR evaluates the form drag coefficient.

CALLING SEQUENCE: FDR = FORMDR (VL)

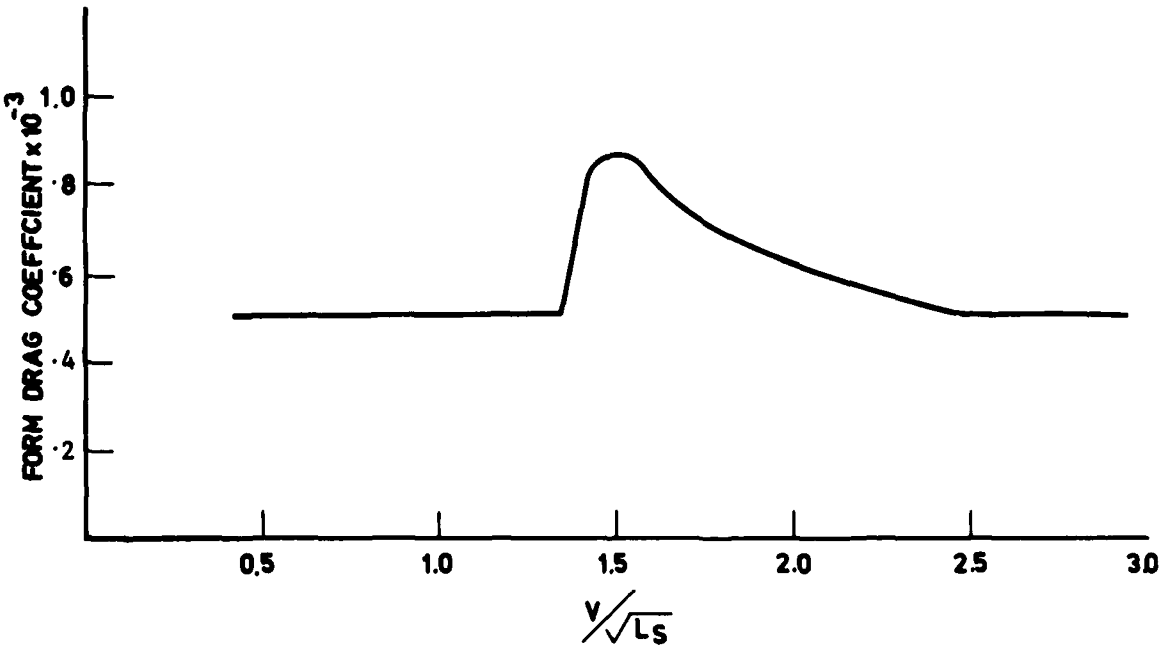
ARGUMENT: VL Ratio of ship speed to square root of length of strut $V/\sqrt{L_S}$

COMMON BLOCKS: NONE

SUBROUTINE CALLED: FUNCTION YINTP

CALLED BY: RWOUT

COMMENTS:



NAME:	SUBROUTINE FRICT
PURPOSE:	Subroutine FRICT calculates the flat plate friction coefficient based on the local Reynolds number.
CALLING SEQUENCE:	CALL FRICT (CF, VF, XLEN, XNU)
ARGUMENTS:	CF Flat plate friction coefficient VF Speed of ship (fps) XLEN Chord of fin (ft) XNU Kinematic viscosity
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	FINDRG
COMMENTS:	

NAME:	SUBROUTINE PCHEB
PURPOSE:	Subroutine PCHEB plots by line printer the body sectional area curve and waterplane outline curve from the given Chebychev coefficients.
CALLING SEQUENCE:	CALL PCHEB (AS, BS, AB, BB, NN, TITLE)
ARGUMENTS:	AS Coefficients of Chebychev Sine Series for strut
	BS Coefficients of Chebychev Cosine Series for strut
	AB Coefficients of Chebychev Sine Series for body
	BB Coefficients of Chebychev Cosine Series for body
	NN Maximum order of Chebychev Series
	TITLE Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCKS:	PLOT, XRPLOTQ
SUBROUTINES CALLED:	PLOT1, PLOT2, PLOT3, PLOT4
CALLED BY:	CHEB
COMMENTS:	

NAME:	SUBROUTINE PLOT1		
PURPOSE:	Subroutine PLOT1 sets up spacing and determines the values of the axes.		
CALLING SEQUENCE:	CALL PLOT 1 (NSCALE, A, B, C, D, E, F)		
ARGUMENTS:	NSCALE		Integer array defined as follows:
	NSCALE (1)	=	I, if printed values of the ordinate are 10 ** I times the actual value
	NSCALE (2)	=	J, if printed values of the ordinate are 10 ** J times the actual value
	NSCALE (3)	=	K, if printed values of the abscissa are 10 ** K times the actual values
	NSCALE (4)	=	L, if printed values of the abscissa are 10 ** L times the actual value
	A		Integer number of horizontal grid lines
	B		Integer number of spaces beyond each horizontal grid line to the next grid line
	C		Integer number of vertical grid lines
	D		Integer number of spaces beyond each vertical grid line to the next grid line
	E		Horizontal grid character
	F		Vertical grid character
COMMON BLOCK:	XRPLOTQ		
SUBROUTINE CALLED:	NONE		
CALLED BY:	PCHEB		
COMMENTS:			

NAME:	SUBROUTINE PLOT2	
PURPOSE:	Subroutine PLOT2 examines the minimum and maximum values of the abscissa and the ordinate and establishes an internal formula for computing location in the image region corresponding to the point to be plotted.	
CALLING SEQUENCE:	CALL PLOT2 (XMAX, XMIN, YMAX, YMIN, NSCLI)	
ARGUMENTS:	XMAX	Value of abscissa at right-most grid line
	XMIN	Value of abscissa at left-most grid line
	YMAX	Value of ordinate at top grid line
	YMIN	Value of ordinate at bottom grid line
	NSCLI	Logical flag (should be false, if PLOT1 has not been called and standard grid is desired)
COMMON BLOCKS:	XRPLOTF, XRPLOTQ, XRPLOTG	
SUBROUTINE CALLED:	NONE	
CALLED BY:	PCHEB	
COMMENTS:		

NAME:	SUBROUTINE PLOT3	
PURPOSE:	Subroutine PLOT3 assigns an alpha-character to each point to be plotted.	
CALLING SEQUENCE:	CALL PLOT3 (PCHAR, X, Y, SDATA, FDATA, DDATA)	
ARGUMENTS:	PCHAR	Plotting character
	X	Array containing the X coordinates to be plotted
	Y	Array containing the Y coordinates to be plotted
	SDATA	Integer position in the arrays of the first ordered pair to be plotted
	FDATA =	1 if each point from SDATA to DDATA is to be plotted 2 if every other point is to be plotted 3 if every third point is to be plotted
	DDATA	Integer position in the array of the last ordered pair to be plotted
COMMON BLOCKS:	XRPLOTF, XRPLOTG	
SUBROUTINE CALLED:	NONE	
CALLED BY:	PCHEB	
COMMENTS:		

NAME:	SUBROUTINE PLOT4	
PURPOSE:	Subroutine PLOT4 prints the image of the completed graph on the printer, including the values of the abscissa and the ordinate at the grid lines outside the bottom and left edge of the graph.	
CALLING SEQUENCE:	CALL PLOT4 (MCHAR, NCHAR)	
ARGUMENTS:	MCHAR	Single dimension array containing alpha characters to be plotted at the left of the graph
	NCHAR	Number of valid characters in MCHAR
COMMON BLOCKS:	XRPLOTF, XRPLOTG, XRPLOTQ	
SUBROUTINE CALLED:	QPLOTZ5	
CALLED BY:	PCHEB	
COMMENTS:		

NAME:	SUBROUTINE POLCOF
PURPOSE:	Subroutine POLCOF determines the coefficients of the polynomials which approximate the KT and KQ curves.
CALLING SEQUENCE:	CALL POLCOF (PD, BAR, Z)
ARGUMENTS:	PD P/D of propeller BAR Blade area ratio Z Number of blades
COMMON BLOCKS:	OFFDES, PCOF
SUBROUTINE CALLED:	Function CALC
CALLED BY:	PROPER
COMMENTS:	

NAME:	SUBROUTINE PRODES
PURPOSE:	Subroutine PRODES is used in the design of a propulsion system to determine characteristics and performance at the design condition.
CALLING SEQUENCE:	CALL PRODES (IS)
ARGUMENT:	IS Type of propulsor (in this program set equal to 1 to indicate conventional fixed blade propellers)
COMMON BLOCKS:	BARCK, CDESC, CPROP
SUBROUTINES CALLED:	TROOST, BURIL
CALLED BY:	SHPCMP
COMMENTS:	

NAME:	SUBROUTINE PROPER	
PURPOSE:	Subroutine PROPER calculates the open water curves' table and off-design performance of propulsor system "IS."	
CALLING SEQUENCE:	CALL PROPER (IS, VOF, TOF, PWR, TITLE)	
ARGUMENTS:	IS	Type of propulsor system (in this program, set to 1 to indicate conventional fixed blade propellers)
	VOF	Off design speed (ft)
	TOF	Off design thrust (lbs)
	PWR	SHP at off design condition
	TITLE	Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCKS:	CDESC, CPROPR, PCOF, ROOTC	
SUBROUTINES CALLED:	POLCOF, ROOTP	
CALLED BY:	SHPCMP	
COMMENTS:		

NAME:	SUBROUTINE PROPP	
PURPOSE:	Subroutine PROPP prints all data in common block "CPROP"	
CALLING SEQUENCE:	CALL PROPP (IS, TITLE)	
ARGUMENTS:	IS	Type of propulsor system (in this program, set to 1 to indicate conventional fixed blade propellers)
	TITLE	Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCK:	CPROP	
SUBROUTINE CALLED:	ERROR	
CALLED BY:	SHPCMP	
COMMENTS:		

NAME:	SUBROUTINE QPLOTZ5
PURPOSE:	Subroutine QPLOTZ5 calculates the scaling information needed to generate the format to label the left-hand side of the program.
CALLING SEQUENCE:	CALL QPLOTZ5 (PDQ)
ARGUMENT:	PDQ Scaling factor for ordinate plot
COMMON BLOCKS:	XRPLOTF, XRPLOTQ
SUBROUTINES CALLED:	NONE
COMMENTS:	

NAME: SUBROUTINE REFLKT

PURPOSE: Subroutine REFLKT reflects the symmetrical matrices to T and W.

CALLING SEQUENCE: CALL REFLKT

ARGUMENTS: NONE

COMMONG BLOCKS: AUX, COEFS

SUBROUTINE CALLED: NONE

CALLED BY: RWAVE

COMMENTS:

NAME:	SUBROUTINE RIN
PURPOSE:	Subroutine RIN reads input data for the program.
CALLING SEQUENCE:	CALL RIN (TITLE)
ARGUMENT:	TITLE Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCK:	INPUT
SUBROUTINE CALLED:	NONE
CALLED BY:	SYNTH
COMMENTS:	

NAME:	SUBROUTINE RINIT
PURPOSE:	Subroutine RINIT initializes the T and W arrays.
CALLING SEQUENCE:	CALL RINIT
ARGUMENTS:	NONE
COMMON BLOCKS:	AUX, COEFS
SUBROUTINE CALLED:	NONE
CALLED BY:	RWAVE
COMMENTS:	

NAME:	SUBROUTINE RINTEG
PURPOSE:	Subroutine RINTEG provides numerical integration for the auxiliary function of T and W.
CALLING SEQUENCE:	CALL RINTEG (ALFA, B, D, NLOC2, WTINT, SEPCOS, SQ)
ARGUMENTS:	ALFA — Integrating variable (α) B — Ratio of distance between centerlines of strut and body to the length of strut D — Ratio of distance between centerlines of second strut and body to the length of strut NLOC2 = $\begin{cases} 0 \text{ if single strut} \\ 1 \text{ if tandem struts} \end{cases}$ WTINT — Weighting constant for numerical integration SEPCOS — Value of the Cosine function in the integrand SQ — Value of $\frac{1}{(\gamma^2 - \gamma^2)^{\frac{1}{2}}}$ in the integrand
COMMON BLOCKS:	AUX, COEFS, INPUT, OMEGA
SUBROUTINE CALLED:	BESSJ
CALLED BY:	RWAVE
COMMENTS:	

NAME:	SUBROUTINE ROOT
PURPOSE:	Subroutine ROOT uses the Regula Falsi method to find the intersection of the "K" curve and the "K/J**2" curve.
CALLING SEQUENCE:	CALL ROOT (PD, Z, BAR, COEFF, IPJ, ITQ, KT, KQ, EFFIN, J2)

ARGUMENTS:

PD	P/D of propeller
Z	Number of blades
BAR	Blade area ratio
COEFF	Optimization coefficients as defined in TROOST
IPJ	Optimization index defined by:
IPJ =	1, no optimization 2, optimize N with KT/J**2 3, optimize N with KQ/J**3 4, optimize D with KT/J**4 5, optimize D with KQ/J**5
ITQ	Index indicating the following:
ITQ =	1, if thrust is input 2, if power is input
KT	Thrust coefficient
KQ	Torque coefficient
EFFIN	Open water efficiency
J2	Advance coefficient

COMMON BLOCK:	ROOTC
SUBROUTINE CALLED:	FUNCTION CALC
CALLED BY:	TROOST
COMMENTS:	

NAME:	SUBROUTINE ROUT
PURPOSE:	Subroutine ROUT outputs the computed results for all components of drag coefficients and effective horsepower.
CALLING SEQUENCE:	CALL ROUT (TITLE, EHP)
ARGUMENTS:	TITLE Array containing the alphanumeric characters of the title of the experiment EHP Effective horsepower of the ship
COMMON BLOCKS:	INPUT, OMEGA, PHYSCO
SUBROUTINES CALLED:	SUM, Function FORMDR
CALLED BY:	SYNTH
COMMENTS:	

NAME: SUBROUTINE RWAVE

PURPOSE: Subroutine RWAVE evaluates the auxiliary functions of T and W.

CALLING SEQUENCE: CALL RWAVE (B, D, NLOC2)

ARGUMENTS:

B	Ratio of distance between center-lines of strut and body to the length of strut
D	Ratio of distance between center-lines of second strut and body to the length of strut
NLOC2	Switch indicating the presence of a second strut

$$\text{NLOC2} = \begin{cases} 0 \text{ if single strut} \\ 1 \text{ if tandem struts} \end{cases}$$

COMMON BLOCKS: AUX, COEFS, INPUT, OMEGA, PHYSCO, PSI

SUBROUTINES CALLED: RINIT, SIMPSN, RINTEG, BESSJ, REFLKT

CALLED BY: SWATH

COMMENTS:

Typical Auxiliary Functions

$$\begin{Bmatrix} \dfrac{T_{Smn}}{(2m-1)(2n-1)} \\ \\ \dfrac{W_{Smn}}{(2m)(2n)} \end{Bmatrix} = \int_{\gamma_{0S}}^{\infty} \frac{d\alpha}{\alpha^2\sqrt{\alpha^2-\gamma_{0S}^2}}\, D\left(\alpha, \frac{2b}{L_S}, \gamma_{0S}\right) \times E_S^2(\alpha) \begin{Bmatrix} J_{2m-1}(\alpha) J_{2n-1}(\alpha) \\ J_{2m}(\alpha) J_{2n}(\alpha) \end{Bmatrix},$$

where

$$D = 1+\cos\left[\left(\frac{2b}{L_S}\right)\left(\frac{2}{\gamma_{0S}}\right)\alpha\sqrt{\alpha^2-\gamma_{0S}^2}\right],$$

$$E_S = 1 - e^{-2(h_S/L_S)(\alpha^2/\gamma_{0S})},$$

$$\gamma_{0S} = \frac{gL_S}{(2U^2)}.$$

NAME: SUBROUTINE SHPCMP

PURPOSE: Subroutine SHPCMP determines a propeller's design and off-design performance and uses this to determine shaft horsepower.

CALLING SEQUENCE: CALL SHPCMP (VDES, EHPDES, PDIA, BDIA, HB, VOFF, EHP, SHP, NV, TITLE)

ARGUMENTS:

VDES	Ship speed at design point (fps)
EHPDES	Effective horsepower at design point
PDIA	Diameter of propeller (ft)
BDIA	Diameter of hull body (ft)
HB	Draft of body (ft)
VOFF	Off-design speed of ship (fps)
EHP	Effective horsepower at off design speeds
SHP	Shaft horsepower at off design speeds
NV	Number of test speeds
TITLE	Array containing the alphanumeric characters of the title of the experiment

COMMON BLOCKS: CDESC, CPROP

SUBROUTINES CALLED: PRODES, PROPP, PROPER, PROPERO

CALLED BY: SYNTH

COMMENTS:

NAME:	SUBROUTINE SIMPSN
PURPOSE:	Subroutine SIMPSN sets up Simpson's multiplier table for numerical integration using Simpson's rule.
CALLING SEQUENCE:	CALL SIMPSN (NPTS, SIMP)
ARGUMENTS:	NPTS — Number of integration steps (odd and greater than one)
	SIMP — Array containing Simpson's coefficients
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	RWAVE
COMMENTS:	

NAME: SUBROUTINE SUM

PURPOSE: Subroutine SUM computes the matrix products as follows:

$$SUM = \sum_{m=1}^{n} \sum_{n=1}^{m} \{A_{sm} A_{sn} T_{smn} + B_{sm} B_{sn} W_{smn}\}$$

CALLING SEQUENCE: CALL SUM (SUM1S, SUM1B, SUM1SB, SUM12S, SUM12B, SUM12SB)

ARGUMENTS:

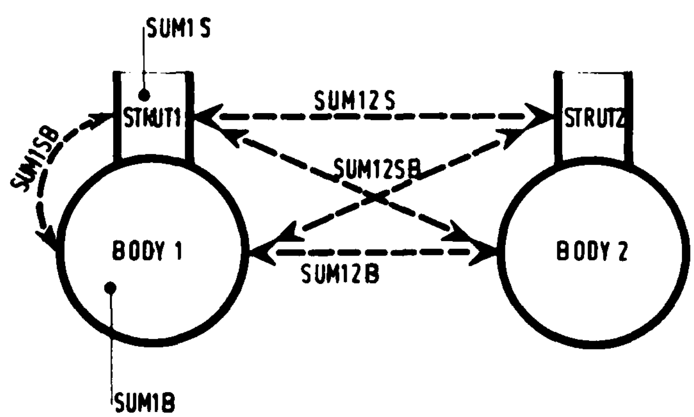
SUM1S	Partial sum for strut 1
SUM1B	Partial sum for body 1
SUM1SB	Partial sum for interaction between strut 1 and body 1
SUBM12S	Partial sum for interaction between strut 1 and strut 2
SUM12B	Partial sum for interaction between body 1 and body 2
SUM12SB	Partial sum for interactions between body 1 and strut 2 or body 2 and strut 1

COMMON BLOCKS: AUX, COEFS

SUBROUTINE CALLED: NONE

CALLED BY: ROUT

COMMENTS:



NAME: SUBROUTINE TROOST

PURPOSE: Subroutine TROOST is used with Subroutine ROOT and Function CALC. Either rpm or diameter can be optimized given the other.

CALLING SEQUENCE: CALL TROOST (VA1, P, T, N1, D, Z, PDNEW, BAR, KT, KQ, EFFNEW, J, ITQ, IPJ, RHO, IERROR)

ARGUMENTS:

VA1	Waked ship velocity (kts)
P	Torque per shaft (ft-lb)
T	Propeller thrust (lbs per shaft)
N1	Propeller rpm
D	Propeller diameter (ft)
Z	Number of blades of propeller
PDNEW	P/D of propeller
BAR	Blade area ratio
KT	Thrust coefficient
KQ	Torque coefficient
EFFNEW	Open water efficiency
J	Advance coefficient
ITQ	Index defined by:
ITQ =	1 if thrust is input 2 if power is input
IPJ	Optimization index defined by:
IPJ =	1 - no optimization, J known, calculates KT, KQ, EFFNEW only 2 - optimize N with KT/J**2 3 - optimize N with KQ/J**3 4 - optimize D with KT/J**4 5 - optimize D with KQ/J**5
RHO	Water density
IERROR	Error message indicator

COMMON BLOCKS: NONE

SUBROUTINES CALLED: ROOT, TROST2, FUNCTION CALC

CALLED BY: PRODES

COMMENTS:

NAME:	SUBROUTINE TROST2
PURPOSE:	Subroutine TROST2 is used to calculate a single KT, KQ and efficiency value for a given propeller.
CALLING SEQUENCE:	CALL TROST2 (KT, KQ, J, PDTR2, BAR, Z, EFFTR2)
ARGUMENTS:	KT Thrust coefficient KQ Torque coefficient J Advance coefficient PDTR2 Pitch to diameter ratio BAR Blade area ratio Z Number of blades EFFTR2 Open water efficiency
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	Function CALC
CALLED BY:	TROOST
COMMENTS:	

NAME:	SUBROUTINE WSURFB
PURPOSE:	Subroutine WSURFB determines the surface area of a body of revolution given by Chebychev coefficients.
CALLING SEQUENCE:	CALL WFSURFB (AREA)
ARGUMENT:	AREA Surface area of a body (ft^2)
COMMON BLOCKS:	COEFS, INPUT, PHYSCO
SUBROUTINE CALLED:	FUNCTION EVAL
CALLED BY:	CHEB
COMMENTS:	

NAME:	SUBROUTINE WSURFS
PURPOSE:	Subroutine WSURFS determines the wetted surface area of a strut.
CALLING SEQUENCE:	CALL WSURFS (AREA)
ARGUMENT:	AREA Wetted surface area of a strut (ft^2)
COMMON BLOCKS:	COEFS, INPUT
SUBROUTINE CALLED:	FUNCTION EVAL
CALLED BY:	CHEB
COMMENTS:	

NAME:	FUNCTION YINTP
PURPOSE:	Function YINTP interpolates through a set of discrete data.
CALLING SEQUENCE:	YTP =YINTP (XA, X, Y, N)
ARGUMENTS:	XA Point to be interpolated X Array of ordinate data Y Array of abscissa data N Number of data points
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	FORMDR
COMMENTS:	

REFERENCES

(1) Lin, L.W. and W.G Day, "The Still-Water Resistance and Propulsion Characteristics of Small-Waterplane-Area Twin-Hull (SWATH) Ships." AIAA/SNAME Advanced Marine Vehicles Conference. San Diego, California, 1974.

(2) Proceedings of the 10th International Towing Tank Conference, London, England, 1963. Published by the National Physical Laboratory, England.

(3) Lasky, M.P., "An Investigation of Appendage Drag." NSRDC/SPD Report 458-H-01.

(4) Roddy, R.F. and J. Strom-Tejsen, "Study of Air Cushion Vehicle Propulsion System - Propulsor Design and Performance." NSRDC/SPD Report 378-09.

APPENDICES

PRECEDING PAGE BLANK-NOT FILMED

PROGRAM LISTING OF CLOSEFIT

```
      PROGRAM SWATH(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8)       SWATH     2
C                                                                      SWATH     3
C     MAIN PROGRAM FOR RESISTANCE OF SWATH SHIP USING CHEBYCHEV        SWATH     4
C     REPRESENTATION OF LINES- REVISED FALL 1974 WINTER 1975 - A.M. REED SWATH   5
C     2-D STRUTS PLUS SUBMERGED ELONGATED BODIES                       SWATH     6
C     INCLUDES 1-2 INTERFERENCE AND MULTIPLE HULL SPACINGS             SWATH     7
C     MERGED WITH CHEB AND REVISED FOR TOTAL RESISTANCE--SPRING 1978   SWATH     8
C                                                                      SWATH     9
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF, SWATH    10
     1           VMFPS,DELCF,TITLE(8),SPACE                            SWATH    11
C                                                                      SWATH    12
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),   SWATH    13
     1           WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10), SWATH    14
     2           WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)           SWATH    15
     3          ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10) SWATH  16
C                                                                      SWATH    17
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,              SWATH    18
     1             SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB, SWATH   19
     2             RATIOL                                              SWATH    20
C                                                                      SWATH    21
      COMMON/PSI/NPTSZ,PTSAF,EXPN,NALMAX,NAL,TAIL,ALFA,ALSMAX,NSTEPS   SWATH    22
C                                                                      SWATH    23
      COMMON / COEFS / ASM(10),BSM(10),ABM(10),BBM(10)                 SWATH    24
C                                                                      SWATH    25
      DATA D, MMAX / 0.0, 10 /                                         SWATH    26
C                                                                      SWATH    27
C                                                                      SWATH    28
C                                                                      SWATH    29
C     XLS    = LENGTH OF STRUT (FT)                                    SWATH    30
C     HS     = DRAFT OF STRUT (FT)                                     SWATH    31
C     TSMAX  = MAXIMUM THICKNESS OF STRUT (FT)                         SWATH    32
C     ASM(I) = STRUT COSINE COEFFICIENTS                               SWATH    33
C     BSM(I) = STRUT SINE COEFFICIENTS                                 SWATH    34
C     XLB    = LENGTH OF BODY (FT)                                     SWATH    35
C     HB     = DEPTH OF SUBMERGENCE TO BODY CENTERLINE (FT)            SWATH    36
C     AX     = MAXIMUM BODY CROSS-SECTIONAL AREA (FT**2)               SWATH    37
C     ABM(I) = BODY COSINE COEFFICIENTS                                SWATH    38
C     BBM(I) = BODY SINE COEFFICIENTS                                  SWATH    39
C     SEPDIS = SEPARATION DISTANCE OF HULL CENTERLINES (FT)            SWATH    40
C     NOSEPS =NUMBER OF SEPARATION DISTANCES TO BE TESTED (FT)         SWATH    41
C     CSTRUT =DISTANCE OF CL OF STRUT FROM CL OF BODY (FT)             SWATH    42
C     CSTRT2 =DISTANCE OF CL OF SECOND STRUT( IF PRESENTED) FROM CL OF SWATH    43
C              BODY (FT)                                               SWATH    44
C              (+ FORWARD OF THE BODY CL, - AFT OF THE BODY CL )       SWATH    45
C     NLOC   =NUMBER OF DIFFERENT STRUT LOCATIONS TO BE TESTED         SWATH    46
C     NLOC2  =FLAG INDICATES 2-D STRUTS                                SWATH    47
C             NLOC2=0, SINGLE STRUT                                    SWATH    48
C             NLOC2 .NE. 0, 2-D STRUTS                                 SWATH    49
C     MMAX   =MAXIMUM NUMBER OF TERMS USED IN CHEBYCHEV APPROXIMATION  SWATH    50
C     WETS   =WETTED SURFACE AREA OF STRUT                             SWATH    51
C     WETB   =WETTED SURFACE AREA OF BODY                              SWATH    52
C     WTSURF =TOTAL WETTED SURFACE AREA OF STRUT AND BODY              SWATH    53
C     SPACE   =DECIMAL FRACTION OF SPACE BETWEEN TANDEM STRUTS TO STRUT SWATH   54
C          --FOR TANDEM STRUTS WHICH ARE NOT EQUAL                     SWATH    55
C                                                                      SWATH    56
C                                                                      SWATH    57
C     INPUT                                                            SWATH    58
```

```
C                                                                         SWATH    59
C                                                                         SWATH    60
  699 CONTINUE                                                            SWATH    61
      READ ( 5,210 ) (TITLE(K),K=1,8)                                     SWATH    62
      READ (5,215) XLS,HS,TSMAX,SPACE                                     SWATH    63
C                                                                         SWATH    64
C     CALL CHEB TO INPUT STRUT DATA AND COMPUTE CHEBYCHEV                 SWATH    65
C     COEFFICIENTS                                                        SWATH    66
C                                                                         SWATH    67
C                                                                         SWATH    68
      CALL CHEB (ASM,BSM,MMAX)                                            SWATH    69
C                                                                         SWATH    70
      READ(5,215) XLB,HB,AX                                               SWATH    71
C                                                                         SWATH    72
C     CALL CHEB TO INPUT BODY DATA AND COMPUTE CHEBYCHEV                  SWATH    73
C     COEFFICIENTS                                                        SWATH    74
C                                                                         SWATH    75
      CALL CHEB (ABM,BBM,MMAX)                                            SWATH    76
C                                                                         SWATH    77
C     COMPUTE TOTAL WETTED SURFACE AREA                                   SWATH    78
C                                                                         SWATH    79
      WTSURF = WETS + WETB                                                SWATH    80
C                                                                         SWATH    81
      READ(5,220) NOSEPS                                                  SWATH    82
      READ(5,215) (SEPDIS(I),I=1,NOSEPS)                                  SWATH    83
      READ(5,220) NLOC,NLOC2                                              SWATH    84
      READ(5,215) (CSTRUT(I),I=1,NLOC)                                    SWATH    85
      IF(NLOC2.NE.0) READ(5,215) (CSTRT2(I),I=1,NLOC)                     SWATH    86
      IF(NLOC2.NE.0) WTSURF=WTSURF+WETS-XLS*TSMAX*ASM(1)*PI/4.0           SWATH    87
C                                                                         SWATH    88
C     ECHO INPUT                                                          SWATH    89
C                                                                         SWATH    90
      CALL ECHO (NLOC)                                                    SWATH    91
C                                                                         SWATH    92
C                                                                         SWATH    93
C     PLOT BODY SECTIONAL AREA CURVE AND STRUT WATERPLANE OUTLINE         SWATH    94
C          CURVE                                                          SWATH    95
C                                                                         SWATH    96
      CALL PCHEB (ASM,BSM,ABM,BBM,MMAX,TITLE)                             SWATH    97
C                                                                         SWATH    98
      HSOLS  = HS/XLS                                                     SWATH    99
      HBOLB  = HB/XLB                                                     SWATH   100
      RATIOL =XLB/XLS                                                     SWATH   101
C                                                                         SWATH   102
C     INPUT TEST VELOCITIES                                               SWATH   103
C                                                                         SWATH   104
C     VMFPS=SHIP SPEED IN FEET PER SECOND                                 SWATH   105
C     JTEST = SWITCH SET TO ZERO IF NO PRINTING OF T AND W                SWATH   106
C             MATRICES IS NEEDED                                          SWATH   107
C                                                                         SWATH   108
  710 READ (5,603) VMFPS, JTEST                                           SWATH   109
      IF (VMFPS.EQ.0.) GO TO 699                                          SWATH   110
      IF(VMFPS.LT.0.0) STOP                                               SWATH   111
      GAMAOS = G*XLS*.5 / VMFPS**2                                        SWATH   112
      GAMAOB = GAMAOS*XLB/XLS                                             SWATH   113
C                                                                         SWATH   114
C     CALCULATE T, W FOR ALL MODEL CONFIGURATION AT ONE                   SWATH   115
```

```
      C     SPEED BY CALLING RWAVE                                        SWATH   116
      C                                                                   SWATH   117
            DO 732 I=1,NLOC                                               SWATH   118
            B=CSTRUT(I)/XLS                                               SWATH   119
            IF(NLOC2.NE.0) D=CSTRT2(I)/XLS                                SWATH   120
            CALL RWAVE(B,D,NLOC2)                                         SWATH   121
            IF (JTEST.EQ.0) GO TO 722                                     SWATH   122
      C                                                                   SWATH   123
      C     PRINT THE AUXILIARY FUNCTION TABLES , IF JTEST .NE. 0         SWATH   124
      C                                                                   SWATH   125
            CALL PRINT (I,NLOC2)                                          SWATH   126
      C                                                                   SWATH   127
      C                                                                   SWATH   128
      C     CALCULATE DRAG COEFFICIENTS AND DRAG FORCES                   SWATH   129
      C                                                                   SWATH   130
      722   CALL RWOUT(I,NLOC2)                                           SWATH   131
      732   CONTINUE                                                      SWATH   132
            GO TO 710                                                     SWATH   133
      C                                                                   SWATH   134
        210 FORMAT(8A10)                                                  SWATH   135
      215   FORMAT(8F10.5)                                                SWATH   136
      220   FORMAT(5I5)                                                   SWATH   137
      603   FORMAT(F10.5,2I5)                                             SWATH   138
            END                                                           SWATH   139
```

```
BLOCK DATA BLKDAT.     74/74    OPT=0 ROUND=*/ TRACE            FTN 4.6+460         04/07/81  13.06.42        PAGE      1

                 BLOCK DATA                                                            BLDATA       2
           C                                                                           BLDATA       3
                 COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,      BLDATA       4
                1           VMFPS,DELCF,TITLE(8),SPACE                                 BLDATA       5
           C                                                                           BLDATA       6
                 COMMON / PSI / NPTSZ,PTSAF,EXPN,NALMAX,NAL,TAIL,ALFA,ALMAX,NSTEPS     BLDATA       7
                 COMMON/PLOT/NFIRST,NLAST,NPOINT,XMAX,XMIN,NSCL1,NCHAR,NSCALE(4),      BLDATA       8
                1            PCHAR(2)                                                  BLDATA       9
                 COMMON/XRPLOTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H      BLDATA      10
           C                                                                           BLDATA      11
                 LOGICAL NSCL1                                                         BLDATA      12
           C                                                                           BLDATA      13
           C        DEFINE PHYSICAL CONSTANTS                                          BLDATA      14
           C                                                                           BLDATA      15
                 DATA PI / 3.1415926535897 /                                           BLDATA      16
                 DATA RHO / 1.9367 /                                                   BLDATA      17
                 DATA G / 32.155 /                                                     BLDATA      18
                 DATA GNU / 1.2970E-5 /                                                BLDATA      19
                 DATA DELCF / 5.0E-4 /                                                 BLDATA      20
           C                                                                           BLDATA      21
           C        DEFINE CONSTANTS FOR NUMERICAL INTEGRATION                         BLDATA      22
           C                                                                           BLDATA      23
                 DATA NPTSZ / 31 /                                                     BLDATA      24
                 DATA PTSAF / 10.0 /                                                   BLDATA      25
                 DATA EXPN / 7.0 /                                                     BLDATA      26
                 DATA NALMAX / 300 /                                                   BLDATA      27
                 DATA ALSMAX / 26.0 /                                                  BLDATA      28
           C                                                                           BLDATA      29
           C        DEFINE DATA FOR PLOT ROUTINE                                       BLDATA      30
           C                                                                           BLDATA      31
           C                                                                           BLDATA      32
                 DATA NHL, NSBH, NVL, NSBV, HCHAR, VCHAR, PCHAR(1), PCHAR(2) /         BLDATA      33
                1      6,   10,   11,   10,   1H.,   1H.,   1HS,    1HB     /          BLDATA      34
           C                                                                           BLDATA      35
                 DATA NFIRST, NLAST, NPOINT / 1, 101, 1 /                              BLDATA      36
           C                                                                           BLDATA      37
                 DATA XMAX, XMIN, NSCL1 / 1.0, -1.0, .TRUE. /                          BLDATA      38
           C                                                                           BLDATA      39
                 DATA NCHAR / 0 /                                                      BLDATA      40
           C                                                                           BLDATA      41
                 DATA NSCALE /0, 3, 0, 3 /                                             BLDATA      42
                 END                                                                   BLDATA      43
```

```
          SUBROUTINE ECHO (NLOC)                                            ECHO      2
C                                                                           ECHO      3
C            ECHO SHIP DATA                                                 ECHO      4
C                                                                           ECHO      5
          COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),    ECHO      6
     1           WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),      ECHO      7
     2           WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)                ECHO      8
     3          ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)    ECHO      9
C                                                                           ECHO     10
          COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,               ECHO     11
     1             SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB,     ECHO     12
     2             RATIOL                                                   ECHO     13
C                                                                           ECHO     14
          COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,  ECHO     15
     1           VMFPS,DELCF,TITLE(8),SPACE                                 ECHO     16
C                                                                           ECHO     17
          COMMON/COEFS/ASM(10),BSM(10),ABM(10),BBM(10)                      ECHO     18
          WRITE (6,310) (TITLE(K),K=1,8)                                    ECHO     19
          WRITE (6,315) MMAX                                                ECHO     20
          WRITE (6,325) XLS,HS,TSMAX                                        ECHO     21
          WRITE (6,330) (ASM(K),K=1,MMAX)                                   ECHO     22
          WRITE (6,335) (BSM(K),K=1,MMAX)                                   ECHO     23
          WRITE (6,340) XLB,HB,AX                                           ECHO     24
          WRITE (6,345) (ABM(K),K=1,MMAX)                                   ECHO     25
          WRITE (6,350) (BBM(K),K=1,MMAX)                                   ECHO     26
          WRITE (6,357) WTSURF                                              ECHO     27
          WRITE (6,355) (SEPDIS(I),I=1,NOSEPS)                              ECHO     28
          WRITE (6,356) (CSTRUT(I),I=1,NLOC)                                ECHO     29
          RETURN                                                            ECHO     30
      310 FORMAT(1H1,///,3X,* SUMMARY OF DATA FOR *,8A10//)                 ECHO     31
      315 FORMAT(1H ,5X,45H NUMBER OF TERMS IN HULL COEFFICIENT SERIES =,I6) ECHO    32
      325 FORMAT(1H ,3X,35H STRUT GEOMETRIC CHARACTERISTICS   ,//,5X,       ECHO     33
     1       8H  XLS  =,F12.6,9X,8H   HS  =,F12.6,9X,8H TSMAX =,F12.6)      ECHO     34
      330 FORMAT(1H0,33X,*STRUT CHEBYCHEV COEFFICIENTS ASM(M)*/(10X,5E16.8)) ECHO    35
      335 FORMAT(1H0,33X,*STRUT CHEBYCHEV COEFFICIENTS BSM(M)*/(10X,5E16.8)) ECHO    36
      340 FORMAT(1H0,3X,35H BODY GEOMETRIC CHARACTERISTICS    ,//,5X,       ECHO     37
     1      8H  XLB  =,F12.6,9X,8H   HB  =,F12.6,9X,8H   AX  =,F12.6)       ECHO     38
      345 FORMAT(1H0,33X,*BODY CHEBYCHEV COEFFICIENTS ABM(M)*/(10X,5E16.8)) ECHO     39
      350 FORMAT(1H0,33X,*BODY CHEBYCHEV COEFFICIENTS BBM(M)*/(10X,5E16.8)) ECHO     40
      355 FORMAT(1H0,5X,* SEPARATION DISTANCE IN FEET     =*,8F8.3)         ECHO     41
      356 FORMAT(*0      STRUT LOCATION IN FEET      =*,8(2X,F10.4))        ECHO     42
      357 FORMAT(1H0,4X,* WETTED SURFACE IN SQUARE FEET  =*,F12.3)          ECHO     43
          END                                                               ECHO     44
```

```
      SUBROUTINE PRINT (I,NLOC2)                                        PRINT      2
C                                                                       PRINT      3
C        PRINT THE AUXILIARY FUNCTION TABLES OF T AND W                 PRINT      4
C                                                                       PRINT      5
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,  PRINT      6
     1            VMFPS,DELCF,TITLE(8),SPACE                            PRINT      7
C                                                                       PRINT      8
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),    PRINT      9
     1            WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10), PRINT     10
     2            WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)           PRINT     11
     3           ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10) PRINT   12
C                                                                       PRINT     13
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,               PRINT     14
     1              SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB, PRINT    15
     2              RATIOL                                              PRINT     16
C                                                                       PRINT     17
C                                                                       PRINT     18
C                                                                       PRINT     19
      WRITE (6,906)                                                     PRINT     20
      WRITE (6,904) CSTRUT(I)                                           PRINT     21
      IF (NLOC2.NE.0) WRITE (6,904) CSTRT2(I)                           PRINT     22
      WRITE (6,903) MMAX,NOSEPS                                         PRINT     23
      WRITE (6,904) GAMAOS,RATIOL,HSOLS,HBOLB,TSMAX,AX,XLS              PRINT     24
      WRITE (6,904) (SEP(J),J=1,NOSEPS)                                 PRINT     25
      WRITE (6,905)                                                     PRINT     26
      WRITE (6,904) ((TS(M,N),M=1,MMAX),N=1,MMAX)                       PRINT     27
      WRITE (6,905)                                                     PRINT     28
      WRITE (6,904) ((WS(M,N),M=1,MMAX),N=1,MMAX)                       PRINT     29
      WRITE (6,905)                                                     PRINT     30
      WRITE (6,904) ((TB(M,N),M=1,MMAX),N=1,MMAX)                       PRINT     31
      WRITE (6,905)                                                     PRINT     32
      WRITE (6,904) ((WB(M,N),M=1,MMAX),N=1,MMAX)                       PRINT     33
      WRITE (6,905)                                                     PRINT     34
      WRITE (6,904) ((TSB (M,N),M=1,MMAX),N=1,MMAX)                     PRINT     35
      WRITE (6,905)                                                     PRINT     36
      WRITE (6,904) ((TSBP(M,N),M=1,MMAX),N=1,MMAX)                     PRINT     37
      WRITE (6,905)                                                     PRINT     38
      WRITE (6,904) ((WSB (M,N),M=1,MMAX),N=1,MMAX)                     PRINT     39
      WRITE (6,905)                                                     PRINT     40
      WRITE (6,904) ((WSBP(M,N),M=1,MMAX),N=1,MMAX)                     PRINT     41
      DO 712 L=1,NOSEPS                                                 PRINT     42
      WRITE (6,906)                                                     PRINT     43
      WRITE (6,904) ((TS12  (L,M,N),M=1,MMAX),N=1,MMAX)                 PRINT     44
      WRITE (6,905)                                                     PRINT     45
      WRITE (6,904) ((WS12  (L,M,N),M=1,MMAX),N=1,MMAX)                 PRINT     46
      WRITE (6,905)                                                     PRINT     47
      WRITE (6,904) ((TB12  (L,M,N),M=1,MMAX),N=1,MMAX)                 PRINT     48
      WRITE (6,905)                                                     PRINT     49
      WRITE (6,904) ((WB12  (L,M,N),M=1,MMAX),N=1,MMAX)                 PRINT     50
      WRITE (6,905)                                                     PRINT     51
      WRITE (6,904) ((TSB12 (L,M,N),M=1,MMAX),N=1,MMAX)                 PRINT     52
      WRITE (6,906)                                                     PRINT     53
      WRITE (6,904) ((TSB12P(L,M,N),M=1,MMAX),N=1,MMAX)                 PRINT     54
      WRITE (6,905)                                                     PRINT     55
      WRITE (6,904) ((WSB12 (L,M,N),M=1,MMAX),N=1,MMAX)                 PRINT     56
      WRITE (6,905)                                                     PRINT     57
      WRITE (6,904) ((WSB12P (L,M,N),M=1,MMAX),N=1,MMAX)                PRINT     58
```

```
  712 CONTINUE
      RETURN
  903 FORMAT(1X,8I10)
  904 FORMAT(1X,10E13.5)
  905 FORMAT(1X)
  906 FORMAT (1H1,30X,* AUXILIARY FUNCTION TABLE OF T AND W *)
      END
```

```
      SUBROUTINE RWOUT(I,NLOC2)
C
C        COMPUTE DRAG COMPONENTS, DRAG COEFFICIENTS, AND TOTAL
C           RESISTANCE
C
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
     1           VMFPS,DELCF,TITLE(8),SPACE
C
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
     1           WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),
     2           WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)
     3           ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)
C
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,
     1             SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB,
     2             RATIOL
C
      DO 88 L=1,NOSEPS
C
C        SUM = A(TRANSPOSE)*T*A + B(TRANSPOSE)*W*B
C
      CALL SUM(L,SUM1S,SUM1B,SUM1SB,SUM12S,SUM12B,SM12SB)
C
      B2OLB = SEPDIS(L)/XLB
      CSOLB=CSTRUT(I)/XLB
C
C     REYNOLDS NUMBER OF STRUT AND BODY
C
      RNS = XLS * VMFPS / GNU
      RNB = XLB * VMFPS / GNU
      IF(NLOC2.NE.0) CSOLB2=CSTRT2(I)/XLB
      AAS = (TSMAX/HS)*(PI/2.0)*GAMAOS
      AAB = 2.0*PI*(AX/XLB**2)*(GAMAOB/GAMAOS**2)
      AASB = (2.0*PI)*AX/(HS*XLS)
      AAWTSF = (0.5*RHO*WTSURF*VMFPS**2)/1000.
C
C     COMPUTE WAVE RESISTANCE
C
      CMS1 = AAS*SUM1S
      R1S = CMS1*RHO*G*TSMAX*HS*XLS
      CWS1    = R1S/AAWTSF
C
      CMB1 = AAB*SUM1B
      R1B = CMB1*RHO*G*AX*XLB
      CWB1    = R1B/AAWTSF
C
      CMSB1 = AASB*SUM1SB
      R1SB = CMSB1*RHO*G*TSMAX*HS*XLS
      CWSB1   = R1SB/AAWTSF
      C1 = CWS1+CWB1+CWSB1
C
      CMS12 = AAS*SUM12S
      R12S = CMS12*RHO*G*TSMAX*HS*XLS
      CWS12   = R12S/AAWTSF
C
      CMB12 = AAB*SUM12B
      R12B = CMB12*RHO*G*AX*XLB
```

```
      CWB12  = R12B/AAWTSF
C
      CMSB12 = AASB*SM12SB
      R12SB = CMSB12*RHO*G*TSMAX*HS*XLS
      CWSB12 = R12SB/AAWTSF
      C2 = CWS12+CWB12+CWSB12
      C3 = C1+C2
C
C     FROUDE NUMBER
      FROUDS = VMFPS/SQRT(32.155*XLS)
      VLS = FROUDS*SQRT(G)/1.6878
      FROUDB = VMFPS/SQRT(32.155*XLB)
      VLB = FROUDB*SQRT(G)/1.6878
      VMKNTS = VMFPS/1.6878
C
C     COMPUTE THE FORM DRAG
C
C
      CFORM = FORMDR (VLS)
      CR = C3 + CFORM
      RR = CR * AAWTSF
C
C     COMPUTE FRICTIONAL DRAG
C
      CFS = CFITTC (RNS)
      CFB = CFITTC (RNB)
      RFS = 0.5*RHO*VMFPS*VMFPS*WETS*(CFS+DELCF)
      RFB = 0.5*RHO*VMFPS*VMFPS*WETB*(CFS+DELCF)
      RF = RFS + RFB
      CFSP = RFS / AAWTSF
      CFBP = RFB / AAWTSF
      CF = CFSP + CFBP
C
C     COMPUTE TOTAL DRAG
C
      CT = CR + CF
      RT = RR + RF
C
C        COMPUTE EHP
C
      EHP = RT * VMFPS / 550.
C
C     OUTPUT
C
      WRITE(6,410) (TITLE(K),K=1,8)
      WRITE(6,325) XLS,HS,TSMAX
      WRITE(6,415)         GAMAOS,FROUDS,HSOLS
      WRITE(6,340) XLB,HB,AX
      WRITE(6,420)         GAMAOB,FROUDB,HBOLB
      WRITE(6,356) SEPDIS(L),B2OLB
      IF(NLOC2.NE.0) GO TO 15
      WRITE(6,358) CSTRUT(I),CSOLB
      GO TO 20
   15 WRITE(6,359) CSTRUT(I),CSOLB,CSTRT2(I),CSOLB2
   20 WRITE(6,357) WTSURF
      WRITE (6,360) VMFPS,VMKNTS,VLS,VLB
      WRITE(6,430) CMS1,R1S,CWS1
```

```
            WRITE(6,440) CMB1,R1B,CWB1
            WRITE(6,445) CMSB1,R1SB,CWSB1,C1
            WRITE(6,450) CMS12,R12S,CWS12
            WRITE(6,455) CMB12,R12B,CWB12
            WRITE(6,460) CMSB12,R12SB,CWSB12,C2,C3
            WRITE (6,510) CR, RR
            WRITE (6,520) CF, RF
            WRITE (6,530) CT, RT
            WRITE (6,540) EHP
   88       CONTINUE
            RETURN
  325 FORMAT(1H ,3X,35H STRUT GEOMETRIC CHARACTERISTICS    ,//,5X,
     1       8H  XLS  =,F12.6,9X,8H   HS  =,F12.6,9X,8H TSMAX =,F12.6)
  340 FORMAT(1H0,3X,35H BODY GEOMETRIC CHARACTERISTICS     ,//,5X,
     1       8H  XLB  =,F12.6,9X,8H   HB  =,F12.6,9X,8H   AX  =,F12.6)
  356       FORMAT(*0      SEPARATION DISTANCE IN FEET     =*,F8.3,
     1      5X,*  B/LB              =*,F7.3)
  357 FORMAT(1H0,4X,* WETTED SURFACE OF A DEMI-HULL IN SQUARE FEET  =*,
     1F12.3)
  358       FORMAT(*0      STRUT CL FROM BODY CL IN FEET    =*,F8.3,
     1     5X,*STRUT OFSET/LB            =*,F7.3)
  359       FORMAT(*0   FORWARD  STRUT CL FROM BODY CL IN FEET   =*,F8.3,
     1     5X,*FORWARD STRUT OFSET/LB            =*,F7.3//
     2       *     AFT  STRUT CL FROM BODY CL IN FEET    =*,F8.3,
     3     5X,*AFT STRUT OFSET/LB            =*,F7.3)
  360 FORMAT(1H0,5X,*  SHIP SPEED IS *,F8.3,* FPS, *,F8.3,* KNOTS, *,
     1       6X,*  STRUT V-L RATIO =*,F8.3,6X,*BODY V-L RATIO = *,F8.3)
  410 FORMAT(*1    *,35H SHIP RESISTANCE CALCULATIONS FOR      ,8A10)
  415 FORMAT(1H ,4X,                8HGAMAOS =,F12.6,9X,8HFROUDS =,
     1       F12.6,9X,8H HS/LS =,F12.6)
  420 FORMAT(1H ,4X,                8HGAMAOB =,F12.6,9X,8HFROUDB =,
     1       F12.6,9X,8H HB/LB =,F12.6)
  430 FORMAT(1H0,30H  WAVE RESISTANCE IN POUNDS                      ,
     1        30H RSW/(RHO*G*TSMAX*HS*LS)     =    ,E16.8,
     2       /,30H   FOR A DEMI-HULL                   ,
     3       32H STRUT WAVE DRAG IN POUNDS    =   ,F10.3,/30X,
     2       32H R1S/(RHO/2*WTSURF*V**2)      =   ,F10.3,4HE-03)
  440 FORMAT(1H0,29X,30H RBW/(RHO*G*AX*LB)              =,E16.8,/30X,
     1       32H BODY WAVE DRAG IN POUNDS     =   ,F10.3,/30X,
     2       32H R1B/(RHO/2*WTSURF*V**2)      =   ,F10.3,4HE-03)
  445 FORMAT(1H0,29X,30H RSBW/(RHO*G*TSMAX*HS*LS)     =,E16.8,/30X,
     1       32H STRUT-BODY WAVE DRAG IN LBS =   ,F10.3,/30X,
     2       32H R1SB/(RHO/2*WTSURF*V**2)     =   ,F10.3,4HE-03,
     3       10X,10H CW1 TOT =,F10.3,4HE-03)
  450 FORMAT(1H0,29X,30H R12SW/(RHO*G*TSMAS*HS*LS)    =,E16.8,/30X,
     1       32H STRUT INTERFERENCE DRAG LBS =   ,F10.3,/30X,
     2       32H R12S/(RHO/2*WTSURF*V**2)     =   ,F10.3,4HE-03)
  455 FORMAT(1H0,29X,30H R12BW/(RHO*G*AX*LB)            =,E16.8,/30X,
     1       32H BODY INTERFERENCE DRAG  LBS =   ,F10.3,/30X,
     2       32H R12B/(RHO/2*WTSURF*V**2)     =   ,F10.3,4HE-03)
  460 FORMAT(1H0,29X,30H R12SBW (RHO*G*TSMAX*HS*LS)  =,E16.8,/30X,
     1       32H STRUT-BODY INTFRNCE DRAG LB =   ,F10.3,/30X,
     2       32H R12SB/(RHO/2*WTSURF*V**2)    =   ,F10.3,4HE-03,
     3       10X,10H CW2 TOT =,F10.3,4HE-03,,/,86X,10H CW  TOT =,F10.3,
     4       4HE-03)
  510 FORMAT(1H0,29HRESIDUAL RESISTANCE          ,
     1       30H  RESIDUAL DRAG COEFFICIENT =  ,F10.3,4HE-03,
```

```
       2      /30X,30H    RESIDUAL RESISTANCE   =     ,F10.3,4H LBS)      RWOUT     173
     520 FORMAT(1H0,29X,30H FRICTIONAL DRAG COEFF.      =,F10.3,4HE-03,    RWOUT     174
       1       /30X,30H FRICTIONAL RESISTANCE      =,F10.3,4H LBS)         RWOUT     175
     530 FORMAT(//13X,*TOTAL DRAG COEFFICIENT IS  *,F10.3,4HE-03,          RWOUT     176
       1       //15X,*TOTAL RESISTANCE FOR A DEMI-HULL *,F10.3,4H LBS)     RWOUT     177
     540 FORMAT (//20X,* EHP FOR A DEMI-HULL *,F10.3)                      RWOUT     178
         END                                                               RWOUT     179
```

```
            SUBROUTINE SIMPSN(NPTS,SIMP)                                SIMPN      2
      C                                                                 SIMPN      3
      C     SET UP SIMPSON*S MULTIPLIERS FOR INTEGRATION                SIMPN      4
      C     ODD NUMBER OF POINTS GREATER THAN 1 REQUIRED                SIMPN      5
      C                                                                 SIMPN      6
            DIMENSION SIMP(NPTS)                                        SIMPN      7
            IF((NPTS/2)*2.NE.NPTS) GO TO 3                              SIMPN      8
            WRITE(5,90)                                                 SIMPN      9
            NPTS =NPTS+1                                                SIMPN     10
      3     SIMP(1) = 1.0                                               SIMPN     11
            SIMP(NPTS) = 1.0                                            SIMPN     12
            NPTSM1=NPTS-1                                               SIMPN     13
            SM=2.0                                                      SIMPN     14
            DO 5 J=2,NPTSM1                                             SIMPN     15
            SM=6.0-SM                                                   SIMPN     16
      5     SIMP(J)=SM                                                  SIMPN     17
            RETURN                                                      SIMPN     18
      90    FORMAT(1H0,*EVEN NO. OF POINTS GIVEN - ADDITIONAL POINT SUPPLIED*) SIMPN 19
            END                                                         SIMPN     20
```

SUBROUTINE SUM   74/74   OPT=0 ROUND=*/ TRACE   FTN 4.6+460   04/07/81 13.06.42   PAGE   1

```
      SUBROUTINE SUM(L,SUM1S,SUM1B,SUM1SB,SUM12S,SUM12B,SM12SB)           SUM       2
C                                                                         SUM       3
C     COMPUTE SUM=A(TRANSPOSE)*T*A+B(TRANSPOSE)*W*B                       SUM       4
C                                                                         SUM       5
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),      SUM       6
     1           WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),    SUM       7
     2           WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)              SUM       8
     3          ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)  SUM       9
C                                                                         SUM      10
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,                 SUM      11
     1             SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB,   SUM      12
     2             RATIOL                                                 SUM      13
C                                                                         SUM      14
      COMMON / COEFS / ASM(10),BSM(10),ABM(10),BBM(10)                    SUM      15
      SUMA1=0.0                                                           SUM      16
      SUMA2=0.0                                                           SUM      17
      SUMA3=0.0                                                           SUM      18
      SUMA4=0.0                                                           SUM      19
      SUMA5=0.0                                                           SUM      20
      SUMA6=0.0                                                           SUM      21
      SUMB1=0.0                                                           SUM      22
      SUMB2=0.0                                                           SUM      23
      SUMB3=0.0                                                           SUM      24
      SUMB4=0.0                                                           SUM      25
      SUMB5=0.0                                                           SUM      26
      SUMB6=0.0                                                           SUM      27
      DO 10 M=1,MMAX                                                      SUM      28
      DO 10 N=1,MMAX                                                      SUM      29
      SUMA1  = SUMA1+ASM(M)*ASM(N)*TS(M,N)                                SUM      30
      SUMB1  = SUMB1+BSM(M)*BSM(N)*WS(M,N)                                SUM      31
      SUMA2  = SUMA2+ABM(M)*ABM(N)*TB(M,N)                                SUM      32
      SUMB2  = SUMB2+BBM(M)*BBM(N)*WB(M,N)                                SUM      33
      SUMA3  = SUMA3+ABM(N)*(ASM(M)*TSB(M,N)+BSM(M)*TSBP(M,N))            SUM      34
      SUMB3  = SUMB3+BBM(N)*(BSM(M)*WSB(M,N)+ASM(M)*WSBP(M,N))            SUM      35
      SUMA4  = SUMA4+ASM(M)*ASM(N)*TS12(L,M,N)                            SUM      36
      SUMB4  = SUMB4+BSM(M)*BSM(N)*WS12(L,M,N)                            SUM      37
      SUMA5  = SUMA5+ABM(M)*ABM(N)*TB12(L,M,N)                            SUM      38
      SUMB5  = SUMB5+BBM(M)*BBM(N)*WB12(L,M,N)                            SUM      39
      SUMA6  = SUMA6+ABM(N)*(ASM(M)*TSB12(L,M,N)+BSM(M)*TSB12P(L,M,N))    SUM      40
      SUMB6  = SUMB6+BBM(N)*(BSM(M)*WSB12(L,M,N)+ASM(M)*WSB12P(L,M,N))    SUM      41
   10 CONTINUE                                                            SUM      42
      SUM1S  = SUMA1+SUMB1                                                SUM      43
      SUM1B  = SUMA2+SUMB2                                                SUM      44
      SUM1SB = SUMA3+SUMB3                                                SUM      45
      SUM12S = SUMA4+SUMB4                                                SUM      46
      SUM12B = SUMA5+SUMB5                                                SUM      47
      SM12SB = SUMA6+SUMB6                                                SUM      48
      RETURN                                                              SUM      49
      END                                                                 SUM      50
```

SYMBOLIC REFERENCE MAP (R=1)

```
      SUBROUTINE RWAVE(B,D,NLOC2)                                       RWAVE    2
C                                                                       RWAVE    3
C        SUBROUTINE FOR COMPUTING AUXILIARY FUNCTIONS T AND W           RWAVE    4
C              USING CHEBYCHEV EXPANSION - PROGRAM VERSION OF 8/05/72   RWAVE    5
C              MODIFICATIONS BY A M REED OCTOBER 1974                   RWAVE    6
C                                                                       RWAVE    7
      DIMENSION VJALFA(21), SIMP(100), SEPCOS(8)                        RWAVE    8
C                                                                       RWAVE    9
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,  RWAVE   10
     1           VMFPS,DELCF,TITLE(8),SPACE                             RWAVE   11
C                                                                       RWAVE   12
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),    RWAVE   13
     1           WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),  RWAVE   14
     2           WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)            RWAVE   15
     3          ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10) RWAVE   16
C                                                                       RWAVE   17
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,               RWAVE   18
     1             SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB, RWAVE   19
     2             RATIOL                                               RWAVE   20
C                                                                       RWAVE   21
C                                                                       RWAVE   22
      COMMON/PSI/NPTSZ,PTSAF,EXPN,NALMAX,NAL,TAIL,ALFA,ALSMAX,NSTEPS    RWAVE   23
C                                                                       RWAVE   24
      DATA PI / 3.1415926535897 /                                       RWAVE   25
C                                                                       RWAVE   26
C        NOSEPS IS THE NUMBER OF HULL SEPARATIONS                       RWAVE   27
C        MMAX IS MAXIMUM NUMBER OF TERMS IN CHEBCHEV SERIES             RWAVE   28
C                                                                       RWAVE   29
      MMAX2 = 2*MMAX                                                    RWAVE   30
      GOSQ=GAMAOS**2                                                    RWAVE   31
      PHIS = 2.0*HSOLS/GAMAOS                                           RWAVE   32
      PHIB = 2.0*HBOLB/GAMAOB                                           RWAVE   33
      RATIOL = GAMAOB/GAMAOS                                            RWAVE   34
C                                                                       RWAVE   35
C        SET THE T AND W ARRAYS TO ZERO                                 RWAVE   36
C                                                                       RWAVE   37
      CALL RINIT                                                        RWAVE   38
      DO 5 L=1,NOSEPS                                                   RWAVE   39
    5 SEP(L) = (2.0/GAMAOS)*SEPDIS(L)/XLS                               RWAVE   40
C                                                                       RWAVE   41
C        INTEGRATION OF T AND W FROM ALPHA EQUALS GAMMAOS TO GAMMAOS+1.0 RWAVE   42
C              MAKE TRIGNOMETRIC SUBSTITUTION TO AVOID SQUAREROOT       RWAVE   43
C              SINGULARITY AT ALPHA EQUALS GAMMAOS                      RWAVE   44
C                                                                       RWAVE   45
C        ZETA IS THE VARIABLE OF INTEGRATION--ZETA=SQRT(ALPHA-GAMMAOS)  RWAVE   46
C        NPTSZ IS THE NUMBER OF  POINTS TO TAKE IN THE INTERVAL         RWAVE   47
C              0.0 LESS THAN OR EQUAL ZETA LESS THAN OR EQUAL 1.0       RWAVE   48
C        AINCR IS THE STEP SIZE FOR THE ZETA INTEGRATION                RWAVE   49
C        ALFAZ IS THE VALUE OF ALPHA FOR ZETA VARYING FROM ZERO TO ONE  RWAVE   50
C                                                                       RWAVE   51
      CALL SIMPSN(NPTSZ,SIMP)                                           RWAVE   52
      XNPTZ=FLOAT(NPTSZ)                                                RWAVE   53
      AINCR = 1.0/(XNPTZ-1.0)                                           RWAVE   54
      HZETA = AINCR/3.0                                                 RWAVE   55
      DO 30 I=1,NPTSZ                                                   RWAVE   56
      SIMP(I) = SIMP(I)*HZETA                                           RWAVE   57
      EI=FLOAT(I-1)                                                     RWAVE   58
```

```
      ZETA = AINCR*EI
      ALFAZ = GAMAOS+ZETA**2
      ALFASQ=ALFAZ**2
      SQZTA=2.0/SQRT(GAMAOS+ALFAZ)
      DO 11 L=1,NOSEPS
   11 SEPCOS(L) = COS(SEP(L)*ALFAZ*SQRT(ALFASQ-GOSQ))
      CALL RINTEG(ALFAZ,B,D,NLOC2,SIMP(I),SEPCOS,SQZTA)
   30 CONTINUE
C
C        INTEGRATION FOR T AND W FROM ALPHA EQUALS GAMMAOS+1.0 TO ALMAX
C
      NAL =  0
      ALFA = GAMAOS + 1.
      PWT = 0.
      ALMAX = ((2.302585*EXPN*GAMAOS) / (2.*HBOLB*RATIOL)
     1                + (GAMAOS+1.)**2 )**.5
  210 ISIMP = 1
      IF (ALFA.GT.ALMAX) GO TO 215
      IF (NAL.LT.NALMAX) GO TO 212
      WRITE (6,302)
      GO TO 215
  212 NAL = NAL + 2
      DQDA = (2.*ALFA**2 - GAMAOS**2)/(ALFA**2-GAMAOS**2)**.5
      SSIZE=(2.*PI)/((2.*RATIOL+DQDA*SEP(1))*PTSAF)
      TWT= SSIZE/3.
      WTINT = TWT + PWT
      PWT = TWT
      GO TO 250
  215 ISIMP=3
      WTINT = PWT
      GO TO 250
  220 WTINT = 4.*TWT
      ISIMP = 2
  250 CONTINUE
      ALFASQ=ALFA**2
      SQ=1.0/SQRT(ALFASQ-GOSQ)
      DO 41 L=1,NOSEPS
   41 SEPCOS(L) = COS(SEP(L)*ALFA/SQ)
      CALL RINTEG(ALFA,B,D,NLOC2,WTINT,SEPCOS,SQ)
      ALFA = ALFA + SSIZE
      IF (ISIMP-2) 220,210,270
C
C        ADD TAIL INTEGRATION TO TS AND WS
C
  270 ALFA = ALFA - SSIZE
      NSTEPS = 0
      IF (ALSMAX.LE.ALFA) GO TO 278
      SSIZE = PI/PTSAF
      ANSTEP = (ALSMAX-ALFA)/SSIZE
      NSTEPS = IFIX(ANSTEP/2.)*2 + 2
      SSIZE = (ALSMAX-ALFA) / NSTEPS
      WT = SSIZE/3.
      ANWT = 1.
      NDO = NSTEPS + 1
      CON=1.0
      DO 277 IDO = 1,NDO
      ALFASQ=ALFA**2
```

```
      IF(NLOC2.NE.0) CON=2.0*(1.0+COS((B-D)*ALFA*2.0))                 RWAVE    116
      CALL BESSJ(ALFA,MMAX2,VJALFA)                                    RWAVE    117
      VS = PHIS*ALFASQ                                                 RWAVE    118
      SQ=1.0/SQRT(ALFASQ-GOSQ)                                         RWAVE    119
      ESA = 1.0                                                        RWAVE    120
      IF ( VS .LT. 300.0 ) ESA = ESA- EXP(-VS)                         RWAVE    121
      ES = SQ * ESA**2 / ALFASQ                                        RWAVE    122
      DO 276 M = 1,MMAX                                                RWAVE    123
      MAA = 2*M                                                        RWAVE    124
      MBB = MAA + 1                                                    RWAVE    125
      DO 276 N = M,MMAX                                                RWAVE    126
      NAA = 2*N                                                        RWAVE    127
      NBB = NAA + 1                                                    RWAVE    128
      WTINT = ANWT * WT                                                RWAVE    129
      TSA = WTINT*ES*VJALFA(MAA)*VJALFA(NAA)*CON                       RWAVE    130
      WSA = WTINT*ES*VJALFA(MBB)*VJALFA(NBB)*CON                       RWAVE    131
      TS(M,N) = TS(M,N) + TSA                                          RWAVE    132
  276 WS(M,N) = WS(M,N) + WSA                                          RWAVE    133
      ALFA = ALFA + SSIZE                                              RWAVE    134
      BNWT = 2.                                                        RWAVE    135
      IF (ANWT.LE.2.) BNWT = 4.                                        RWAVE    136
      IF (IDO.EQ.NDO-1) BNWT = 1.                                      RWAVE    137
  277 ANWT = BNWT                                                      RWAVE    138
  278 ALFASQ=ALFA**2                                                   RWAVE    139
      RAT = GAMAOS/ALFA                                                RWAVE    140
      IF (RAT.LT.1.E-4) GO TO 280                                      RWAVE    141
      TAIL=(-SQRT(ALFASQ-GOSQ)/ALFASQ+ASIN(RAT)/GAMAOS)/(2.0*PI*GOSQ)  RWAVE    142
      GO TO 290                                                        RWAVE    143
  280 TAIL=1.0/(3.0*PI*ALFA*ALFASQ)                                    RWAVE    144
C                                                                      RWAVE    145
C        CALCULATE TAIL RESULTS                                        RWAVE    146
C                                                                      RWAVE    147
  290 DO 296 M = 1,MMAX                                                RWAVE    148
      DO 296 N = M,MMAX                                                RWAVE    149
      SIGN = (-1)**(M+N)                                               RWAVE    150
      TS(M,N) = TS(M,N) - SIGN*TAIL                                    RWAVE    151
      WS(M,N) = WS(M,N) + SIGN*TAIL                                    RWAVE    152
  296 CONTINUE                                                         RWAVE    153
C                                                                      RWAVE    154
C        MULTIPLY BY CONSTANTS AND REFLECT SYMMETRIC MATRICIES)        RWAVE    155
C                                                                      RWAVE    156
      CALL REFLKT                                                      RWAVE    157
      RETURN                                                           RWAVE    158
  302 FORMAT (*1ALFA INTEGRATION REACHED NALMAX*)                      RWAVE    159
      END                                                              RWAVE    160
```

```
      SUBROUTINE RINIT                                                  RINIT    2
C                                                                       RINIT    3
C        INITIALIZE THE T AND W ARRAYS TO ZERO                          RINIT    4
C                                                                       RINIT    5
C                                                                       RINIT    6
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),    RINIT    7
     1           WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),  RINIT    8
     2          WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)             RINIT    9
     3         ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10) RINIT   10
C                                                                       RINIT   11
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,               RINIT   12
     1             SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB, RINIT   13
     2             RATIOL                                               RINIT   14
C                                                                       RINIT   15
C                                                                       RINIT   16
      DO 4 M=1,MMAX                                                     RINIT   17
      DO 4 N=M,MMAX                                                     RINIT   18
      TS(M,N)  = 0.0                                                    RINIT   19
      WS(M,N)  = 0.0                                                    RINIT   20
      TB(M,N)  = 0.0                                                    RINIT   21
      WB(M,N)  = 0.0                                                    RINIT   22
      TSB(M,N) = 0.0                                                    RINIT   23
      TSBP(M,N) = 0.0                                                   RINIT   24
      TSB(N,M)=0.0                                                      RINIT   25
      TSBP(N,M)=0.0                                                     RINIT   26
      WSB(M,N) = 0.0                                                    RINIT   27
      WSBP(M,N) = 0.0                                                   RINIT   28
      WSB(N,M)=0.0                                                      RINIT   29
      WSBP(N,M)=0.0                                                     RINIT   30
      DO 4 L=1,NOSEPS                                                   RINIT   31
      TS12(L,M,N) = 0.0                                                 RINIT   32
      WS12(L,M,N) = 0.0                                                 RINIT   33
      TB12(L,M,N) = 0.0                                                 RINIT   34
      WB12(L,M,N) = 0.0                                                 RINIT   35
      TSB12(L,M,N) = 0.0                                                RINIT   36
      TSB12P(L,M,N) = 0.0                                               RINIT   37
      TSB12(L,N,M)=0.0                                                  RINIT   38
      TSB12P(L,N,M)=0.0                                                 RINIT   39
      WSB12(L,M,N) = 0.0                                                RINIT   40
      WSB12P(L,M,N) = 0.0                                               RINIT   41
      WSB12(L,N,M)=0.0                                                  RINIT   42
      WSB12P(L,N,M)=0.0                                                 RINIT   43
    4 CONTINUE                                                          RINIT   44
      RETURN                                                            RINIT   45
      END                                                               RINIT   46
```

```
      SUBROUTINE RINTEG(ALFA,B,D,NLOC2,WTINT,SEPCOS,SQ)                  RINTG     2
C                                                                        RINTG     3
C        NUMERICAL INTEGRATION FOR AUXILIARY FUNCTIONS T AND W           RINTG     4
C                                                                        RINTG     5
C                                                                        RINTG     6
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),     RINTG     7
     1           WSB(10.10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),   RINTG     8
     2           WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)             RINTG     9
     3          ,TSBP(10,10).WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10) RINTG    10
C                                                                        RINTG    11
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,                RINTG    12
     1              SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB, RINTG    13
     2              RATIOL                                               RINTG    14
C                                                                        RINTG    15
C                                                                        RINTG    16
      DIMENSION VJALFA(21), VJBETA(21), SEPCOS(8)                        RINTG    17
C                                                                        RINTG    18
      MMAX2=2*MMAX                                                       RINTG    19
      BETA = RATIOL*ALFA                                                 RINTG    20
      ALFASQ=ALFA**2                                                     RINTG    21
C                                                                        RINTG    22
C        CALCULATE FACTORS FOR STRUT NOT CENTERED                        RINTG    23
C                                                                        RINTG    24
      CO=COS(B*ALFA*2.0)                                                 RINTG    25
      SI=SIN(B*ALFA*2.0)                                                 RINTG    26
      CON=1.0                                                            RINTG    27
      IF (NLOC2.EQ.0) GO TO 32                                           RINTG    28
C                                                                        RINTG    29
C        CALCULATE FACTORS FOR TANDEM STRUTS                             RINTG    30
C                                                                        RINTG    31
      CO=(CO+COS(D*ALFA*2.0))                                            RINTG    32
      SI=(SI+SIN(D*ALFA*2.0))                                            RINTG    33
      CON=2.0*(1.0+COS((B-D)*ALFA*2.0))                                  RINTG    34
   32 CONTINUE                                                           RINTG    35
C                                                                        RINTG    36
C        COMPUTE AUXILIARY FUNCTIONS VJALFA AND VJBETA                   RINTG    37
C                                                                        RINTG    38
      CALL BESSJ(ALFA,MMAX2,VJALFA)                                      RINTG    39
      CALL BESSJ(BETA,MMAX2,VJBETA)                                      RINTG    40
C                                                                        RINTG    41
C        COMPUTE AUXILIARY FUNCTIONS ES AND EB                           RINTG    42
C                                                                        RINTG    43
      VS=PHIS*ALFASQ                                                     RINTG    44
      VB = PHIB*BETA**2                                                  RINTG    45
      ESA = 1.0                                                          RINTG    46
      EBA = 0.0                                                          RINTG    47
      IF(VS .LT. 300.) ESA = ESA-EXP(-VS)                                RINTG    48
      IF(VB .LT. 300.) EBA = EXP(-VB)                                    RINTG    49
      ES=SQ*ESA**2/ALFASQ                                                RINTG    50
      EB=SQ*EBA**2*ALFASQ                                                RINTG    51
      ESB = SQ*ESA*EBA                                                   RINTG    52
C                                                                        RINTG    53
C        TB,TS,WB,WS,TW12,TS12,WB12, AND WS12 ARE ALL SYMMETRIC WRT M    RINTG    54
C              AND N SO WE NEED ONLY CALCULATE THE UPPER HALF OF THESE   RINTG    55
C              MATRICIES                                                 RINTG    56
C                                                                        RINTG    57
      DO 55 M=1,MMAX                                                     RINTG    58
```

```
      MAA = 2*M                                                          RINTG    59
      MBB = 2*M+1                                                        RINTG    60
      DO 55 N=M,MMAX                                                     RINTG    61
      NAA = 2*N                                                          RINTG    62
      NBB = 2*N+1                                                        RINTG    63
      TSA =   WTINT*ES*VJALFA(MAA)*VJALFA(NAA)*CON                       RINTG    64
      WSA =   WTINT*ES*VJALFA(MBB)*VJALFA(NBB)*CON                       RINTG    65
      IF(EB .EQ. 0.0) GO TO 54                                           RINTG    66
      TBA =   WTINT*EB*VJBETA(MAA)*VJBETA(NAA)                           RINTG    67
      WBA =   WTINT*EB*VJBETA(MBB)*VJBETA(NBB)                           RINTG    68
      TSBA =    WTINT*ESB*VJBETA(NAA)*VJALFA(MAA)*CO                     RINTG    69
      TSBAP= - WTINT*ESB*VJBETA(NAA)*VJALFA(MBB)*SI                      RINTG    70
      WSBA =    WTINT*ESB*VJBETA(NBB)*VJALFA(MBB)*CO                     RINTG    71
      WSBAP=    WTINT*ESB*VJBETA(NBB)*VJALFA(MAA)*SI                     RINTG    72
      TSBB =    WTINT*ESB*VJBETA(MAA)*VJALFA(NAA)*CO                     RINTG    73
      TSBBP= - WTINT*ESB*VJBETA(MAA)*VJALFA(NBB)*SI                      RINTG    74
      WSBB =    WTINT*ESB*VJBETA(MBB)*VJALFA(NBB)*CO                     RINTG    75
      WSBBP=    WTINT*ESB*VJBETA(MBB)*VJALFA(NAA)*SI                     RINTG    76
      TB(M,N) = TB(M,N)+TBA                                              RINTG    77
      WB(M,N) = WB(M,N)+WBA                                              RINTG    78
      TSB(M,N) = TSB(M,N)+TSBA                                           RINTG    79
      TSBP(M,N) = TSBP(M,N)+TSBAP                                        RINTG    80
      WSB(M,N) = WSB(M,N)+WSBA                                           RINTG    81
      WSBP(M,N) = WSBP(M,N)+WSBAP                                        RINTG    82
      IF(M.EQ.N) GO TO 54                                                RINTG    83
      TSB(N,M) = TSB(N,M)+TSBB                                           RINTG    84
      TSBP(N,M) = TSBP(N,M)+TSBBP                                        RINTG    85
      WSB(N,M) = WSB(N,M)+WSBB                                           RINTG    86
      WSBP(N,M) = WSBP(N,M)+WSBBP                                        RINTG    87
   54 CONTINUE                                                           RINTG    88
      TS(M,N) = TS(M,N)+TSA                                              RINTG    89
      WS(M,N) = WS(M,N)+WSA                                              RINTG    90
      DO 55 L=1,NOSEPS                                                   RINTG    91
      TS12(L,M,N) = TS12(L,M,N)+TSA*SEPCOS(L)                            RINTG    92
      WS12(L,M,N) = WS12(L,M,N)+WSA*SEPCOS(L)                            RINTG    93
      IF(EBA .EQ. 0.0) GO TO 55                                          RINTG    94
      TB12(L,M,N) = TB12(L,M,N)+TBA*SEPCOS(L)                            RINTG    95
      WB12(L,M,N) = WB12(L,M,N)+WBA*SEPCOS(L)                            RINTG    96
      TSB12(L,M,N) = TSB12(L,M,N)+TSBA*SEPCOS(L)                         RINTG    97
      TSB12P(L,M,N) = TSB12P(L,M,N)+TSBAP*SEPCOS(L)                      RINTG    98
      WSB12(L,M,N) = WSB12(L,M,N)+WSBA*SEPCOS(L)                         RINTG    99
      WSB12P(L,M,N) = WSB12P(L,M,N)+WSBAP*SEPCOS(L)                      RINTG   100
      IF(M.EQ.N) GO TO 55                                                RINTG   101
      TSB12(L,N,M) = TSB12(L,N,M)+TSBB*SEPCOS(L)                         RINTG   102
      TSB12P(L,N,M) = TSB12P(L,N,M)+TSBBP*SEPCOS(L)                      RINTG   103
      WSB12(L,N,M) = WSB12(L,N,M)+WSBB*SEPCOS(L)                         RINTG   104
      WSB12P(L,N,M) = WSB12P(L,N,M)+WSBBP*SEPCOS(L)                      RINTG   105
   55 CONTINUE                                                           RINTG   106
      RETURN                                                             RINTG   107
      END                                                                RINTG   108
```

```
      SUBROUTINE REFLKT
C
C        REFLECT THE SYMMETRICAL MATRICES OF T AND W
C
C
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
     1           WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),
     2           WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)
     3          ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)
C
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOB,NOSEPS,
     1              SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB,
     2              RATIOL
C
C
      DO 63 M=1,MMAX
      EM=FLOAT(M)
      U = 2.0*EM-1.0
      DO 63 N=M,MMAX
      EN=FLOAT(N)
      FOURMN=4.0*EM*EN
      UV=U*(2.0*EN-1.0)
      TS(M,N) = TS(M,N)*UV
      TS(N,M) = TS(M,N)
      WS(M,N) = WS(M,N)*FOURMN
      WS(N,M) = WS(M,N)
      TB(M,N) = TB(M,N)*UV
      TB(N,M) = TB(M,N)
      WB(M,N) = WB(M,N)*FOURMN
      WB(N,M) = WB(M,N)
      TSB(M,N) = TSB(M,N)*UV
      TSBP(M,N) = TSBP(M,N)*UV
      WSB(M,N) = WSB(M,N)*FOURMN
      WSBP(M,N) = WSBP(M,N)*FOURMN
      IF (M.EQ.N) GO TO 62
      TSB(N,M)=TSB(N,M)*UV
      TSBP(N,M)=TSBP(N,M)*UV
      WSB(N,M)=WSB(N,M)*FOURMN
      WSBP(N,M)=WSBP(N,M)*FOURMN
   62 CONTINUE
      DO 63 L=1,NOSEPS
      TS12(L,M,N) = TS12(L,M,N)*UV
      TS12(L,N,M) = TS12(L,M,N)
      WS12(L,M,N) = WS12(L,M,N)*FOURMN
      WS12(L,N,M) = WS12(L,M,N)
      TB12(L,M,N) = TB12(L,M,N)*UV
      TB12(L,N,M) = TB12(L,M,N)
      WB12(L,M,N) = WB12(L,M,N)*FOURMN
      WB12(L,N,M) = WB12(L,M,N)
      TSB12(L,M,N) = TSB12(L,M,N)*UV
      TSB12P(L,M,N) = TSB12P(L,M,N)*UV
      WSB12(L,M,N) = WSB12(L,M,N)*FOURMN
      WSB12P(L,M,N) = WSB12P(L,M,N)*FOURMN
      IF(M.EQ.N) GO TO 63
      TSB12(L,N,M)=TSB12(L,N,M)*UV
      TSB12P(L,N,M)=TSB12P(L,N,M)*UV
      WSB12(L,N,M)=WSB12(L,N,M)*FOURMN
```

```
             WSB12P(L,N,M)=WSB12P(L,N,M)*FOURMN                          REFLK      59
      63     CONTINUE                                                    REFLK      60
             RETURN                                                      REFLK      61
             END                                                         REFLK      62
```

```
      SUBROUTINE BESSJ(X,N,VJ)                                  BESSJ       2
C                                                               BESSJ       3
C     EVALUATE THE VALUES OF BESSEL FUNCTION FROM ORDER 0 TO    BESSJ       4
C     ORDER N WITH ARGUMENT X                                   BESSJ       5
C                                                               BESSJ       6
      REAL J, JL, JLL, JLLL                                     BESSJ       7
      DIMENSION VJ(N)                                           BESSJ       8
      DATA PI /3.1415926535898/,                                BESSJ       9
     1     EPS /1.0E-30/,                                       BESSJ      10
     2     EXPON /2.7182818284590/                              BESSJ      11
C                                                               BESSJ      12
C        DETERMINE THE MAXIMUM N = NU                           BESSJ      13
C                                                               BESSJ      14
      D = ALOG(EPS)/X                                           BESSJ      15
      C = ALOG(EXPON/2.0)                                       BESSJ      16
      B = D-2.0-C                                               BESSJ      17
      ALPHA = (-B+SQRT(B**2-4.0*D*(2.0-C)))/(2.0*(2.0-C))       BESSJ      18
      CLN = ALOG(2.0*PI*X)                                      BESSJ      19
      IF(ALPHA.LT.1.5) ALPHA = 1.5                              BESSJ      20
      DO 1000 I=1,20                                            BESSJ      21
         ALN = ALOG(ALPHA)                                      BESSJ      22
         F = -(CLN+ALN)/(2.0*X)+ALPHA*(C-ALN)-D                 BESSJ      23
         FP = -1.0/(2.0*X*ALPHA)+C-1.0-ALN                      BESSJ      24
         DEL = F/FP                                             BESSJ      25
      ALPHA = ALPHA-DEL                                         BESSJ      26
         IF(X*ABS(DEL) .LE. 0.01) GO TO 1100                    BESSJ      27
 1000 CONTINUE                                                  BESSJ      28
      GO TO 88888                                               BESSJ      29
 1100 CONTINUE                                                  BESSJ      30
      GNU = ALPHA*X                                             BESSJ      31
      NU = IFIX(GNU)+1                                          BESSJ      32
      IF(NU.LT.N+10) NU = N+10                                  BESSJ      33
      IF((NU/2)*2.NE.NU) NU=NU+1                                BESSJ      34
C                                                               BESSJ      35
C         RECURSE IN PLACE FOR M.GT.N                           BESSJ      36
C                                                               BESSJ      37
      SUM = 0.0                                                 BESSJ      38
      JLLL = 0.0                                                BESSJ      39
      JLL = EPS                                                 BESSJ      40
      NUM1 = NU-1                                               BESSJ      41
      NP = N                                                    BESSJ      42
      IF((NP/2)*2.EQ.NP) NP = NP-1                              BESSJ      43
      DO 1200 MM = NP,NUM1,2                                    BESSJ      44
         M = NUM1-MM+NP                                         BESSJ      45
         JL = 2.0*FLOAT(M+2)*JLL/X-JLLL                         BESSJ      46
         J = 2.0*FLOAT(M+1)*JL/X-JLL                            BESSJ      47
         SUM = SUM+2.0*JL                                       BESSJ      48
         JLLL = JL                                              BESSJ      49
         JLL = J                                                BESSJ      50
 1200 CONTINUE                                                  BESSJ      51
      IF(NP.NE.N) GO TO 1300                                    BESSJ      52
      JL = 2.0*FLOAT(N)*JLL/X-JLLL                              BESSJ      53
      SUM = SUM+2.0*JL                                          BESSJ      54
      JLLL = JLL                                                BESSJ      55
      JLL = JL                                                  BESSJ      56
 1300 CONTINUE                                                  BESSJ      57
      I = N+1                                                   BESSJ      58
```

```
            VJ(I) = JLLL                                                BESSJ     59
            VJ(N) = JLL                                                 BESSJ     60
            NP = N-1                                                    BESSJ     61
            DO 1400 MM = 1,NP                                           BESSJ     62
               M = NP-MM                                                BESSJ     63
               K = M+1                                                  BESSJ     64
               VJ(K) = 2.0*FLOAT(K)*VJ(M+2)/X-VJ(M+3)                   BESSJ     65
               IF((M/2)*2.EQ.M) SUM = SUM+2*VJ(M+1)                     BESSJ     66
      1400  CONTINUE                                                    BESSJ     67
            SUM = SUM-VJ(1)                                             BESSJ     68
            K = N+1                                                     BESSJ     69
            DO 1500 M = 1,K                                             BESSJ     70
               VJ(M) = VJ(M)/SUM                                        BESSJ     71
      1500  CONTINUE                                                    BESSJ     72
            RETURN                                                      BESSJ     73
     88888  CONTINUE                                                    BESSJ     74
            WRITE(6,100) X, ALPHA, DEL                                  BESSJ     75
            STOP                                                        BESSJ     76
       100  FORMAT(28H NO CONVERGENCE, X,ALPHA,DEL/3E14.7)              BESSJ     77
            END                                                         BESSJ     78
```

```
      SUBROUTINE PCHEB (AS, BS, AB, BB, NN, TITLE)
C
C     PLOT BODY SECTIONAL AREA CURVE AND STRUT WATERPLANE OUTLINE CURVE
C     INPUTS
C         AS AND BS ARE, RESPECTIVELY, THE VECTORS OF SYMMETRIC-
C         AND OF ANTISYMMETRIC-CHEBYCHEV COEFFICIENTS FOR THE STRUT.
C
C         AB AND BB ARE, RESPECTIVELY, THE VECTORS OF SYMMETRIC-
C         AND OF ANTISYMMETRIC-CHEBYCHEV COEFFICIENTS FOR THE BODY.
C
C         NN = DIMENSION OF AS, BS, AB, BB
C           MAXIMUM VALUE OF NN IS NMAX
C           NMAX = 10
C
C
C
C
C     INITIALIZE THE GRID GEOMETRY, THE PLOT VARIABLES AND
C     THE ABSCISSA VARIABLE.
C
      COMMON/PLOT/NFIRST,NLAST,NPOINT,XMAY,XMIN,NSCL1,NCHAR,NSCALE(4),
     1            PCHAR(2)
      COMMON/XRPLOTQ/II,JJ,KK,LL,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,
     1               ISX,ISY,V,H
C
      LOGICAL NSCL1
C
      DIMENSION AS(NN), BS(NN), AB(NN), BB(NN), TITLE(8)
      DIMENSION STAT(101),YS(101),YB(101),LABLE(3)
C
C
C    GENERATE ORDINATE VALUES FOR STRUT AND BODY POINTS
C
      DO 2000 I=1, 101
         STAT(I)=2.0*(FLOAT(I-1)*.01-.5)
         THETA=ASIN(STAT(I))
         YS(I)=0.0
         YB(I)=0.0
         DO 1000 NCO=1,NN
            UM=COS(FLOAT(2*NCO-1)*THETA)
            VM=SIN(FLOAT(2*NCO)*THETA)
            YS(I)=YS(I)+AS(NCO)*UM+BS(NCO)*VM
            YB(I)=YB(I)+AB(NCO)*UM+BB(NCO)*VM
 1000    CONTINUE
 2000 CONTINUE
C
C     DETERMINE THE MINIMUM AND MAXIMUM ORDINATE VALUES
C
      YMAX = -999999.0
      YMIN = +999999.0
      DO 3000 J=1,101
         IF(YMAX .LT. YB(J)) YMAX=YB(J)
         IF(YMAX .LT. YS(J)) YMAX=YS(J)
         IF(YMIN .GT. YB(J)) YMIN=YB(J)
         IF(YMIN .GT. YS(J)) YMIN=YS(J)
 3000 CONTINUE
C
```

```
C     CALL PLOT1 TO SET UP GRID SPACING AND DETERMINE THE AXIS VALUES    PCHEB    59
C                                                                        PCHEB    60
      CALL PLOT1 (NSCALE,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR)                  PCHEB    61
C                                                                        PCHEB    62
C     CALL PLOT2 TO EXAMINE THE MINIMUM AND MAXIMUM VALUES OF            PCHEB    63
C     ABSCISSA AND ORDINATE AND TO ESTABLISH AN INTERNAL FORMULA         PCHEB    64
C     FOR COMPUTING THE LOCATION IN THE IMAGE REGION CORRESPONDING       PCHEB    65
C     TO THE POINT TO BE PLOTTED.                                        PCHEB    66
C     WHEN NSCL1=.TRUE., THE STANDARD GRID WILL NOT BE USED              PCHEB    67
C                                                                        PCHEB    68
      CALL PLOT2 (XMAX,XMIN,YMAX,YMIN,NSCL1)                             PCHEB    69
C                                                                        PCHEB    70
C     CALL PLOT3 FOR ASSIGNING AN ALPHA-CHARACTER TO EACH POINT          PCHEB    71
C     WHICH WILL BE PLOTTED.                                             PCHEB    72
C                                                                        PCHEB    73
      CALL PLOT3 (PCHAR(1),STAT,YS,NFIRST,NLAST,NPOINT)                  PCHEB    74
      CALL PLOT3 (PCHAR(2),STAT,YB,NFIRST,NLAST,NPOINT)                  PCHEB    75
C                                                                        PCHEB    76
C     PRINT THE HEADING OF THE PLOT                                      PCHEB    77
C                                                                        PCHEB    78
      WRITE (6,9000)  (TITLE(I), I=1,8)                                  PCHEB    79
C                                                                        PCHEB    80
C     CALL PLOT4 TO PRINT THE IMAGE OF THE COMPLETED GRAPH ON THE        PCHEB    81
C     PRINTER, INCLUDING THE VALUES OF ABSCISSA AND ORDINATE AT          PCHEB    82
C     THE GRID LINES OUTSIDE THE BOTTOM AND LEFT EDGE OF THE GRAPH       PCHEB    83
C                                                                        PCHEB    84
      CALL PLOT4 (LABLE,NCHAR)                                           PCHEB    85
C                                                                        PCHEB    86
      RETURN                                                             PCHEB    87
C                                                                        PCHEB    88
C                                                                        PCHEB    89
 9000 FORMAT(1H1,32X,64HBODY SECTIONAL AREA CURVE AND STRUT WATERPLANE O PCHEB    90
     1UTLINE CURVE FOR//25X,8A10)                                        PCHEB    91
      END                                                                PCHEB    92
```

```
      SUBROUTINE PLOT1 (NSCALE,A,B,C,D,E,F)                                PLOT1      2
C                                                                          PLOT1      3
C     SETUP SPACING AND DETERMINE THE AXIS VALUES                          PLOT1      4
C                                                                          PLOT1      5
C                                                                          PLOT1      6
      INTEGER A,B,C,D                                                      PLOT1      7
      LOGICAL V,H                                                          PLOT1      8
C                                                                          PLOT1      9
      DIMENSION NSCALE(4)                                                  PLOT1     10
C                                                                          PLOT1     11
      COMMON/XRPLOTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H     PLOT1     12
C                                                                          PLOT1     13
      I=NSCALE(1)                                                          PLOT1     14
      J=NSCALE(2)                                                          PLOT1     15
      K=NSCALE(3)                                                          PLOT1     16
      L=NSCALE(4)                                                          PLOT1     17
      NHL=A                                                                PLOT1     18
      NSBH=B-1                                                             PLOT1     19
      NVL=C                                                                PLOT1     20
      NSBV=D-1                                                             PLOT1     21
      HCHAR=E                                                              PLOT1     22
      VCHAR=F                                                              PLOT1     23
      RETURN                                                               PLOT1     24
      END                                                                  PLOT1     25
```

```
      SUBROUTINE PLOT2(XMAX,XMIN,YMAX,YMIN,NSCL1)
C
C     EXAMINE THE MINIMUM AND MAXIMUM VALUES OF ABSCISSA AND ORDINATE
C     AND ESTABLISH AN INTERNAL FORMULA FOR COMPUTING THE LOCATION
C     IN THE IMAGE REGION CORRESPONDING TO THE POINT TO BE PLOTTED
C
C
      REAL LIN
      INTEGER VCT,VLCT,HCT,HLCT
      LOGICAL NSCL1,V,H
C
      DIMENSION LIN(110)
C
      COMMON/XRPLOTF/XL,XH,YL,YH,XI,YI,XMOV,YMOV
      COMMON/XRPLOTG/GRAF(11,204)
      COMMON/XRPLOTQ/II,JJ,KK,LL,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,IX,IY,V,H
C
      IF(NSCL1) GOTO 1
         II=0
         JJ=3
         KK=0
         LL=3
         NHL=6
         NVL=11
         NSBH=9
         NSBV=9
         HCHAR=1L-
         VCHAR=1LI
    1 CONTINUE
      HLCT=0
      HCT=NSBH
      DO 2 I=1,204
         IF(HLCT.EQ.NHL) GOTO 3
         IY=I
         VLCT=0
         VCT=NSBV
         DO 4 J=1,110
            IF(VLCT.EQ.NVL) GO TO 5
            IX=J
            IF(VCT.EQ.NSBV) GOTO 6
              LIN(J)=1L
            GOTO 40
    6       CONTINUE
               LIN(J)=VCHAR
               VCT=-1
               VLCT=VLCT+1
   40       CONTINUE
            VCT=VCT+1
    4    CONTINUE
    5    CONTINUE
         VLCT=0
         IF(HCT.NE.NSBH) GOTO 7
            HCT=-1
            HLCT=HLCT+1
            DO 8 J=1,IX
               LIN(J)=HCHAR
    8       CONTINUE
```

```
      7    CONTINUE                                                       PLOT2     59
           HCT=HCT+1                                                      PLOT2     60
           IQ=IX+1                                                        PLOT2     61
           DO 10 J=IQ,110                                                 PLOT2     62
              LIN(J)=1L                                                   PLOT2     63
     10    CONTINUE                                                       PLOT2     64
           ENCODE(110,9,GRAF(1,I)) (LIN(J),J=1,110)                       PLOT2     65
      2 CONTINUE                                                          PLOT2     66
      3 CONTINUE                                                          PLOT2     67
        XL=XMIN                                                           PLOT2     68
        XH=XMAX                                                           PLOT2     69
        YL=YMIN                                                           PLOT2     70
        YH=YMAX                                                           PLOT2     71
        XI=(XH-XL)/FLOAT(IX-1)                                            PLOT2     72
        YI=(YH-YL)/FLOAT(IY-1)                                            PLOT2     73
        XMOV=1.0-XL/XI                                                    PLOT2     74
        YMOV=1.0-YL/YI                                                    PLOT2     75
        V=.TRUE.                                                          PLOT2     76
        H=.TRUE.                                                          PLOT2     77
        RETURN                                                            PLOT2     78
      9 FORMAT(110A1)                                                     PLOT2     79
        END                                                               PLOT2     80
```

```
      SUBROUTINE PLOT3 (PCHAR,X,Y,SDATA,FDATA,DDATA)                    PLOT3      2
C                                                                       PLOT3      3
C     ASSIGN AN ALPHA-CHARACTER TO EACH POINT WHICH WILL BE PLOTTED     PLOT3      4
C                                                                       PLOT3      5
C                                                                       PLOT3      6
      REAL LIN                                                          PLOT3      7
      INTEGER ROW, COL                                                  PLOT3      8
      INTEGER SDATA, FDATA, DDATA                                       PLOT3      9
C                                                                       PLOT3     10
      DIMENSION LIN(110)                                                PLOT3     11
      DIMENSION X(FDATA), Y(FDATA)                                      PLOT3     12
C                                                                       PLOT3     13
      COMMON/XRPLOTF/XL,XH,YL,YH,XI,YI,XMOV,YMOV                        PLOT3     14
      COMMON/XRPLOTG/GRAF(11,204)                                       PLOT3     15
C                                                                       PLOT3     16
      DO 18 I = SDATA, FDATA, DDATA                                     PLOT3     17
         IF (Y(I).LT.YL.OR.Y(I).GT.YH) GOTO 16                          PLOT3     18
            ROW=IFIX(Y(I)/YI+YMOV)                                      PLOT3     19
            IF(ROW.LT.1.OR.ROW.GT.204) GOTO 14                          PLOT3     20
               DECODE(110,2,GRAF(1,ROW)) (LIN(J),J=1,110)               PLOT3     21
               IF(X(I).LT.XL.OR.X(I).GT.XH) GOTO 12                     PLOT3     22
                  COL=IFIX(X(I)/XI+XMOV)                                PLOT3     23
                  IF(COL.LT.1.OR.COL.GT.110) GOTO 10                    PLOT3     24
                     LIN(COL)=PCHAR                                     PLOT3     25
                     ENCODE(110,2,GRAF(1,ROW)) (LIN(J),J=1,110)         PLOT3     26
   10             CONTINUE                                              PLOT3     27
   12          CONTINUE                                                 PLOT3     28
   14       CONTINUE                                                    PLOT3     29
   16    CONTINUE                                                       PLOT3     30
   18 CONTINUE                                                          PLOT3     31
      RETURN                                                            PLOT3     32
    2 FORMAT(110A1)                                                     PLOT3     33
      END                                                               PLOT3     34
```

```
      SUBROUTINE PLOT4(MCHAR,NCHAR)
C
C     PRINT THE IMAGE OF THE COMPLETED GRAPH ON THE PRINTER,
C     INCLUDING THE VALUES OF THE ABSCISSA AND ORDINATE AT THE
C     GRID LINES OUTSIDE THE BOTTOM AND LEFT EDGES OF THE GRAPH
C
C
      INTEGER SA, SB, SC
      INTEGER CT,PDQ
      LOGICAL V,H
C
      DIMENSION HNUM(15), MCHAR(3), LCHAR(30), VGFMT(3), HLFMT(3)
C
      COMMON/XRPLOTF/XL,XH,YL,YH,XI,YI,XMOV,YMOV
      COMMON/XRPLOTG/GRAF(11,204)
      COMMON/XRPLOTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H
C
      SA=NSBV
      SC=NCHAR
      SB=NVL
      KQ=1
      CALL QPLOTZ5(PDQ)
      ENCODE(26,2,VGFMT(1))J
      IF(NSBV.LE.0) NSBV=10
    4 CONTINUE
      IF(NSBV.GT. 10) GO TO 3
         NSBV=NSBV*2
         XVL=NVL
         NVL=XVL/2.0+.5
         KQ=KQ+1
         GO TO 4
    3 CONTINUE
      NNN=NVL-1
      MMM=NSBV-8
      ENCODE(25,5,HLFMT(1)) PDQ,L,NNN,MMM,L
      IF (NCHAR.NE.0) GOTO 14
         NCHAR=1
         LCHAR(1)=1L
      GO TO 15
   14 CONTINUE
         DECODE(10,17,MCHAR(1)) (LCHAR(IJ),IJ=1,10)
         DECODE(10,17,MCHAR(2)) (LCHAR(IJ),IJ=11,20)
         DECODE(10,17,MCHAR(3)) (LCHAR(IJ),IJ=21,30)
   15 CONTINUE
      MCT=0
      NSPC=(ISY-NCHAR)/2
      LCT=0
      CT=NSBH
      DO 6 MMNX=1,ISY
         N=ISY-MMNX + 1
         IF (MMNX.LE.NSPC.OR.LCT.GE.NCHAR) GO TO 7
            LCT=LCT+1
            LCR=LCHAR(LCT)
         GO TO 8
    7   CONTINUE
            LCR=1L
    8   CONTINUE
```

```
          IF(CT.EQ.NSBH.AND.H) GO TO 9                                  PLOT4     59
             PRINT 1,LCR,(GRAF(II,N),II=1,11)                           PLOT4     60
          GO TO 13                                                      PLOT4     61
    9   CONTINUE                                                        PLOT4     62
             CT=-1                                                      PLOT4     63
             QNN=(YH-MCT*YI)*10.**I                                     PLOT4     64
             PRINT VGFMT,LCR,ONN,(GRAF(II,N),II=1,11)                   PLOT4     65
             MCT=MCT+NSBH+1                                             PLOT4     66
   13   CONTINUE                                                        PLOT4     67
          CT=CT+1                                                       PLOT4     68
    6 CONTINUE                                                          PLOT4     69
      MCT=0                                                             PLOT4     70
      IF(.NOT.V) GO TO 12                                               PLOT4     71
          DO 10 N=1,NVL                                                 PLOT4     72
             HNUM(N)=(XL+MCT*XI)*10.**K                                 PLOT4     73
             MCT=MCT+NSBV+KQ                                            PLOT4     74
   10     CONTINUE                                                      PLOT4     75
          PRINT 11                                                      PLOT4     76
          PRINT HLFMT,(HNUM(N),N=1,NVL)                                 PLOT4     77
   12 CONTINUE                                                          PLOT4     78
      NSBV=SA                                                           PLOT4     79
      NVL=SB                                                            PLOT4     80
      NCHAR=SC                                                          PLOT4     81
      RETURN                                                            PLOT4     82
    1 FORMAT(1H,A1,16X,11A10)                                           PLOT4     83
    2 FORMAT(15H(1H ,A1,3X,F10.,I1,10H,3X,11A10))                       PLOT4     84
    5 FORMAT(1H(,I2,6HX,F10.,I1,1H.,I2,1H(,I2,6HX,F10.,I1,2H)))         PLOT4     85
   11 FORMAT (1H0)                                                      PLOT4     86
   17 FORMAT(10A1)                                                      PLOT4     87
      END                                                               PLOT4     88
```

```
      SUBROUTINE QPLOTZ5(PDQ)                                    QPLOT    2
C                                                                QPLOT    3
      INTEGER A,B,C,D,PDQ                                        QPLOT    4
      LOGICAL V,H                                                QPLOT    5
C                                                                QPLOT    6
      COMMON/XRPLOTF/XL,XH,YL,YH,XI,YI,XM,YM                     QPLOT    7
      COMMON/XRPLOTQ/II,JJ,KK,LL,A,B,C,D,E,F,M,N,V,H             QPLOT    8
C                                                                QPLOT    9
      X=ABS(XL)                                                  QPLOT   10
      IF(X.LT.XH) X=XH                                           QPLOT   11
      K=0                                                        QPLOT   12
      DO 1 I=1,11                                                QPLOT   13
         XJ=10**(I-1)                                            QPLOT   14
         IF(X.GT.XJ) GOTO 2                                      QPLOT   15
         K=K+1                                                   QPLOT   16
    1 CONTINUE                                                   QPLOT   17
    2 CONTINUE                                                   QPLOT   18
      PDQ=1                                                      QPLOT   19
      IF(LL.EQ.0) PDQ=0                                          QPLOT   20
      QQQ=PDQ+KK+LL+K                                            QPLOT   21
      PDQ=14.0-(10.0-QQQ)/2.0                                    QPLOT   22
      RETURN                                                     QPLOT   23
      END                                                        QPLOT   24
```

```
      SUBROUTINE CHEB (AMC,BMC,MMAX)                                    CHEB   2
C                                                                       CHEB   3
C     INPUT STRUT OR BODY DATA, COMPUTE CHEBYCHEV COEFFICIENTS          CHEB   4
C     CHECK THE ACCURACY OF CHEBYCHEV APPROXIMATION.                    CHEB   5
C                                                                       CHEB   6
      DIMENSION AMC(10), BMC(10)                                        CHEB   7
C                                                                       CHEB   8
      COMMON/NAME/INCHAR(8),NAME(7)                                     CHEB   9
C                                                                       CHEB  10
      COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV        CHEB  11
      LOGICAL ISBODY                                                    CHEB  12
C                                                                       CHEB  13
C                                                                       CHEB  14
C     READ INPUT OFFSET DATA                                            CHEB  15
C                                                                       CHEB  16
      READ (5,31) INCHAR                                                CHEB  17
      IF (INCHAR(1).EQ.5HSTRUT) ISBODY=.FALSE.                          CHEB  18
      IF (INCHAR(1).EQ. 4HBODY) ISBODY=.TRUE.                           CHEB  19
C                                                                       CHEB  20
      CALL READ                                                         CHEB  21
C                                                                       CHEB  22
C     COMPUTE SCALING FACTOR                                            CHEB  23
C                                                                       CHEB  24
C                                                                       CHEB  25
      CALL SCALE(ISBODY)                                                CHEB  26
C                                                                       CHEB  27
C     COMPUTE CHEBYCHEV COEFFICIENTS                                    CHEB  28
C                                                                       CHEB  29
      CALL COMPUT (AMC,BMC,MMAX,ISBODY)                                 CHEB  30
C                                                                       CHEB  31
C     CHECK INPUT OFFSET AGAINST OFFSET OBTAINED BY EVALUATING          CHEB  32
C     THE CHEBYCHEV SERIES                                              CHEB  33
C                                                                       CHEB  34
      CALL CHECK (AMC,BMC,MMAX,ISBODY)                                  CHEB  35
C                                                                       CHEB  36
C     COMPUTE WETTED SURFACE AREA                                       CHEB  37
C                                                                       CHEB  38
C                                                                       CHEB  39
      CALL SURFACE (AMC,BMC,MMAX,ISBODY)                                CHEB  40
C                                                                       CHEB  41
C                                                                       CHEB  42
C         CHECK NUMBER OF OFFSET INPUT CARDS                            CHEB  43
C                                                                       CHEB  44
      READ (5,32) NINPUT                                                CHEB  45
      NINPUT = NINPUT + NNV - 1                                         CHEB  46
      IF (NINPUT.EQ.N) GO TO 999                                        CHEB  47
C                                                                       CHEB  48
C     ERROR IN INPUT CARDS                                              CHEB  49
C                                                                       CHEB  50
      WRITE (6,200) INCHAR                                              CHEB  51
      STOP                                                              CHEB  52
  999 CONTINUE                                                          CHEB  53
      RETURN                                                            CHEB  54
31    FORMAT (8A10)                                                     CHEB  55
   32 FORMAT (I3)                                                       CHEB  56
  200 FORMAT (*  ERROR IN INPUT DATA CARDS ----*,8A10)                  CHEB  57
      END                                                               CHEB  58
```

```
      SUBROUTINE WTSURFB (AMC,BMC,MMAX)                                 WSURB     2
C                                                                       WSURB     3
C        DETERMINE THE SURFACE AREA OF A BODY OF                        WSURB     4
C           REVOLUTION GIVEN BY CHEBYCHEV COEFFICIENTS                  WSURB     5
C                                                                       WSURB     6
      DIMENSION AMC(10),BMC(10)                                         WSURB     7
C                                                                       WSURB     8
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,  WSURB     9
     1           VMFPS,DELCF,TITLE(8),SPACE                             WSURB    10
C                                                                       WSURB    11
      RAD=SQRT(AX/PI)                                                   WSURB    12
      CON=(2.0*RAD/XLB)**2                                              WSURB    13
      DX=0.05                                                           WSURB    14
      DXSQ=DX*DX                                                        WSURB    15
      R=0.0                                                             WSURB    16
      WETB=0.0                                                          WSURB    17
      DO 10 I=1,40                                                      WSURB    18
         RL=R                                                           WSURB    19
         X=DX*FLOAT(I)-1.0                                              WSURB    20
         CALL EVAL (X,MMAX,R,AMC,BMC)                                   WSURB    21
      IF (R.LT.0.0) R=0.0                                               WSURB    22
         R=SQRT(R)                                                      WSURB    23
         RM=(RL+R)*0.5                                                  WSURB    24
         DR=R-RL                                                        WSURB    25
         WETB=WETB+RM*SQRT(DXSQ+CON*DR**2)                              WSURB    26
   10 CONTINUE                                                          WSURB    27
      WETB=PI*RAD*XLB*WETB                                              WSURB    28
      RETURN                                                            WSURB    29
      END                                                               WSURB    30
```

```
      SUBROUTINE WTSURFS (AMC,BMC,MMAX)                                 WSURS    2
C                                                                       WSURS    3
C         DETERMINE THE SURFACE AREA OF A STRUT WHOSE THICKNESS         WSURS    4
C                DISTRIBUTION IS GIVEN BY CHEBYCHEV COEFFICIENTS        WSURS    5
C                                                                       WSURS    6
      DIMENSION AMC(10),BMC(10)                                         WSURS    7
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,  WSURS    8
     1          VMFPS,DELCF,TITLE(8),SPACE                              WSURS    9
C                                                                       WSURS   10
      CON=(TSMAX/XLS)**2                                                WSURS   11
      DX=0.05                                                           WSURS   12
      DXSQ=DX*DX                                                        WSURS   13
      WETS=0.0                                                          WSURS   14
      T=0.0                                                             WSURS   15
      DO 10 I=1,40                                                      WSURS   16
         TL=T                                                           WSURS   17
         X=DX*FLOAT(I)-1.0                                              WSURS   18
         CALL EVAL (X,MMAX,T,AMC,BMC)                                   WSURS   19
         DT=T-TL                                                        WSURS   20
         WETS=WETS+SQRT(DXSQ+CON*DT**2)                                 WSURS   21
   10 CONTINUE                                                          WSURS   22
      WETS = WETS - SPACE * 2.0                                         WSURS   23
      WETS=XLS*HS*WETS                                                  WSURS   24
      RETURN                                                            WSURS   25
      END                                                               WSURS   26
```

```
                 SUBROUTINE READ                                              READ        2
          C                                                                   READ        3
          C        READ OFFSET DATA OF STRUT OR BODY AND CHECK FOR ERROR      READ        4
          C                                                                   READ        5
                COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV    READ        6
                COMMON/NAME/INCHAR(8),NAME(7)                                 READ        7
                COMMON/NORM/YMAX,XDIVER,YDIVER,XADDER                         READ        8
                LOGICAL REPEAT                                                READ        9
                YMAX = 0.                                                     READ       10
                DO 47 I = 2,8                                                 READ       11
          47    NAME(I-1) = INCHAR(I)                                         READ       12
                REPEAT = .FALSE.                                              READ       13
                NNV = 1                                                       READ       14
                DO 40 I = 1,100                                               READ       15
                N=I                                                           READ       16
          C                                                                   READ       17
          C         N = NUMBER OF INPUT POINTS                                READ       18
          C                                                                   READ       19
                IF (REPEAT) GO TO 339                                         READ       20
                READ(5,12) X(I), Y(I), IEND                                   READ       21
                IF (I.EQ.1) GO TO 39                                          READ       22
                IF (I.GT.2) GO TO 38                                          READ       23
                SIGN = X(I) - XLAST                                           READ       24
                IF (SIGN.EQ.0.) GO TO 45                                      READ       25
                GO TO 39                                                      READ       26
          38    IF ((X(I)-XLAST)*SIGN .LE. 0.) GO TO 45                       READ       27
          39    XLAST = X(I)                                                  READ       28
                IF (Y(I).GT.YMAX) YMAX = Y(I)                                 READ       29
                IF (IEND.EQ.5H  END) GO TO 42                                 READ       30
                IF (IEND.NE.5HBREAK) GO TO 40                                 READ       31
                REPEAT = .TRUE.                                               READ       32
                NV(NNV) = I                                                   READ       33
                NNV = NNV + 1                                                 READ       34
                GO TO 40                                                      READ       35
          339   X(I) = X(I-1)                                                 READ       36
                Y(I) = Y(I-1)                                                 READ       37
                REPEAT = .FALSE.                                              READ       38
          40    CONTINUE                                                      READ       39
                WRITE(6,18) NAME                                              READ       40
                STOP                                                          READ       41
             42 CONTINUE                                                      READ       42
                NV(NNV) = N                                                   READ       43
                GO TO 8                                                       READ       44
             45 CONTINUE                                                      READ       45
                PRINT 46, X(I),Y(I)  ,NAME                                    READ       46
                STOP                                                          READ       47
              8 CONTINUE                                                      READ       48
                RETURN                                                        READ       49
          12    FORMAT (2F10.0, 55X, A5)                                      READ       50
          18    FORMAT (* TOO MANY INPUT CARDS FOR   *,7A10)                  READ       51
          46    FORMAT (* CARD OUT OF ORDER *,2E16.8/20X,*FOR-- *,7A10)       READ       52
                END                                                           READ       53
```

```
             SUBROUTINE SCALE (ISBODY)                                       SCALE      2
      C                                                                      SCALE      3
      C        COMPUTE SCALING FACTORS AND SCALE THE INPUT FOR LENGTH        SCALE      4
      C        BETWEEN -1.0 AND +1.0 AND MAXIMUM BEAM OR SECTIONAL           SCALE      5
      C        AREA OF 1.0                                                   SCALE      6
      C                                                                      SCALE      7
             COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV      SCALE      8
             COMMON/NAME/INCHAR(8),NAME(7)                                   SCALE      9
             COMMON/NORM/YMAX,XDIVER,YDIVER,XADDER                           SCALE     10
             LOGICAL ISBODY                                                  SCALE     11
      C                                                                      SCALE     12
      C                                                                      SCALE     13
      54     YDIVER = YMAX                                                   SCALE     14
             XDIVER = (X(N)-X(1))*.5                                         SCALE     15
             XADDER = -1. - X(1) / XDIVER                                    SCALE     16
             IF (ISBODY) WRITE (6,17)                                        SCALE     17
             IF (.NOT.ISBODY) WRITE (6,18)                                   SCALE     18
      55     WRITE(6,19) NAME                                                SCALE     19
             WRITE(6,25)                                                     SCALE     20
             WRITE(6,26) (X(I),Y(I),I=1,N)                                   SCALE     21
      C                                                                      SCALE     22
      C      IF *YDIVER* IS NOT ZERO, THE Y INPUTS WILL BE DIVIDED BY *YDIVER*  SCALE  23
      C          .... SAME FOR *XDIVER* .....                                SCALE     24
      C      THEN *XADDER* IS ADDED TO X INPUTS                              SCALE     25
      C      THEN IF *ISBODY*, THE Y INPUTS WILL BE SQUARED                  SCALE     26
      C                                                                      SCALE     27
             WRITE(6,206) XDIVER,XADDER,YDIVER                               SCALE     28
             IF(ISBODY) WRITE(6,207)                                         SCALE     29
             RETURN                                                          SCALE     30
          17 FORMAT (1H1,* INPUT OFFSET DATA OF BODY *)                      SCALE     31
          18 FORMAT (1H1,* INPUT OFFSET DATA OF STRUT *)                     SCALE     32
          19 FORMAT(1H0,7A10,//)                                             SCALE     33
      25     FORMAT (7X,*X-VALUES*,10X,*Y-VALUES*,10X,*BEFORE SCALING*//)    SCALE     34
      26     FORMAT (2(6X,F10.5))                                            SCALE     35
      206    FORMAT (*0 SCALING--*/*  X VALUES ARE DIVIDED BY *,E16.8,       SCALE     36
          1        *, THEN ADD *,E16.8 / *  Y VALUES ARE DIVIDED BY *,E16.8)  SCALE    37
      207    FORMAT (15X,*AND SQUARED*)                                      SCALE     38
             END                                                             SCALE     39
```

```
      SUBROUTINE COMPUT (AMC,BMC,MMAX,ISBODY)
C
C        COMPUTE THE COEFFICIENTS OF THE CHEBYCHEV POLYNOMIALS
C
      DIMENSION AMC(10),BMC(10)
      COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV
      COMMON/NAME/INCHAR(8),NAME(7)
      COMMON/NORM/YMAX,XDIVER,YDIVER,XADDER
      LOGICAL ISBODY
C
      DO 100 I=1,N
      Y(I) = Y(I) / YDIVER
      IF (ISBODY) Y(I) = Y(I)**2
      XWAS = X(I)
      X(I) = X(I)/XDIVER + XADDER
      IF (ABS(X(I)) .LE. 1.) GO TO 90
      IF (ABS(X(I)) .GT. 1.01) GO TO 90
      WRITE(6,201) X(I),XWAS
      X(I) = AINT(X(I))
   90 THETA(I) = ASIN(X(I))
  100 CONTINUE
      IF (ISBODY) WRITE (6,13)
      IF(.NOT.ISBODY) WRITE (6,14)
      WRITE(6,19) NAME
      WRITE(6,15)
      WRITE(6,16)(X(I),Y(I),THETA(I),I=1,N)
C
C         COMPUTE FOLLOWS
C
      CALL CHEV (AMC,BMC,MMAX,ISBODY)
C
      RETURN
   13 FORMAT(1H1,* NORMALIZED OFFSET DATA FOR BODY *)
   14 FORMAT(1H1,* NORMALIZED OFFSET DATA FOR STRUT *)
   15 FORMAT (7X,*X-VALUES*,10X,*Y-VALUES*,10X,*  THETA *//)
   16 FORMAT (3(6X,F10.5))
   19 FORMAT(1H0,7A10,//)
  201 FORMAT (*0TRUNCATED X= *,2E16.8)
      END
```

```
      SUBROUTINE SURFACE (AMC,BMC,MMAX,ISBODY)                          SURFC    2
C                                                                       SURFC    3
C        FIND THE WETTED SURFACE AREA OF STRUT OR BODY. ALSO THE        SURFC    4
C           WATER PLANE AREA OF STRUT AND DISPLACED VOLUME OF BODY      SURFC    5
C                                                                       SURFC    6
      DIMENSION AMC(10),BMC(10)                                         SURFC    7
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,  SURFC    8
     1             VMFPS,DELCF,TITLE(8),SPACE                           SURFC    9
      COMMON/NAME/INCHAR(8),NAME(7)                                     SURFC   10
      LOGICAL ISBODY                                                    SURFC   11
      DATA PI / 3.1415926535897 /                                       SURFC   12
C                                                                       SURFC   13
      IF (ISBODY) GO TO 100                                             SURFC   14
C                                                                       SURFC   15
C        COMPUTE WETTED SURFACE AREA OF STRUT                           SURFC   16
C                                                                       SURFC   17
      CALL WTSURFS (AMC,BMC,MMAX)                                       SURFC   18
C                                                                       SURFC   19
C        COMPUTE WATER PLANE AREA OF STRUT                              SURFC   20
C                                                                       SURFC   21
      WPA = XLS * TSMAX * AMC(1) * PI / 4.0                             SURFC   22
      WRITE (6,110) WETS,XLS,TSMAX,WPA                                  SURFC   23
      GOTO 999                                                          SURFC   24
  100 CONTINUE                                                          SURFC   25
C                                                                       SURFC   26
C        COMPUTE WETTED SURFACE AREA OF BODY                            SURFC   27
C                                                                       SURFC   28
      CALL WTSURFB (AMC,BMC,MMAX)                                       SURFC   29
      WETB = WETB - WPA                                                 SURFC   30
C                                                                       SURFC   31
C        COMPUTE DISPLACED VOLUME                                       SURFC   32
C                                                                       SURFC   33
      DISPVOL=XLB*AX*PI/4.0*AMC(1)                                      SURFC   34
      WRITE (6,120) WETB,XLB,AX,DISPVOL                                 SURFC   35
  999 CONTINUE                                                          SURFC   36
      RETURN                                                            SURFC   37
  110 FORMAT (1H0,*WETTED SURFACE AREA OF STRUT IS *,E16.8,             SURFC   38
     1      *,WHERE STRUT LENGTH IS *,E16.8,                            SURFC   39
     2      *   AND MAXIMUM THICKNESS IS *,E16.8,                       SURFC   40
     3      /,20X,*WATER PLANE AREA IS *,E16.8)                         SURFC   41
  120 FORMAT(1H0,*WETTED SURFACE AREA OF BODY IS *,E16.8,               SURFC   42
     1      *,WHERE BODY LENGTH IS *,E16.8,                             SURFC   43
     2      * AND MAXIMUM AREA IS *,E16.8/                              SURFC   44
     3      20X,* BODY VOLUME IS *,E16.8)                               SURFC   45
      END                                                               SURFC   46
```

```
              SUBROUTINE CHEV (AMC,BMC,MMAX,ISBODY)                          CHEV    2
      C                                                                      CHEV    3
      C          THIS PROGRAM USES THE OFFSETS TO COMPUTE CHEBYCHEV          CHEV    4
      C          COEFFICIENTS VIA SPLINE APPROXIMATION                       CHEV    5
      C                                                                      CHEV    6
              DIMENSION AMC(10),BMC(10)                                      CHEV    7
              COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV     CHEV    8
              COMMON/NAME/INCHAR(8),NAME(7)                                  CHEV    9
              LOGICAL ISBODY                                                 CHEV   10
              DATA PI / 3.1415926535897 /                                    CHEV   11
      C                                                                      CHEV   12
      C       OBTAIN CURVE FIT OF OFFSETS                                    CHEV   13
      C                                                                      CHEV   14
              ISTART = 1                                                     CHEV   15
              DO 50 I = 1,NNV                                                CHEV   16
              NN = NV(I)                                                     CHEV   17
              NPTS = NN-ISTART+1                                             CHEV   18
              CALL SPLINE(THETA(ISTART),Y(ISTART),D(ISTART),NPTS,BC,1,1,ISBODY) CHEV   19
      50      ISTART = NN + 1                                                CHEV   20
              N = NV(NNV)                                                    CHEV   21
      C                                                                      CHEV   22
      C       AMC AND BMC ARE THE COEFFICIENTS THAT ARE CALCULATED AND STORED CHEV   23
      C                                                                      CHEV   24
      C       MAIN CALCULATION FOLLOWS                                       CHEV   25
      C                                                                      CHEV   26
              DO 15 J =1,MMAX                                                CHEV   27
              AJ=FLOAT(J)                                                    CHEV   28
              ARCU = (2.0*AJ-1.0)                                            CHEV   29
              ARCV = 2.0*AJ                                                  CHEV   30
              AMC(J) = SUMSPL(THETA,Y,D,N,ARCU,-1)                           CHEV   31
              BMC(J) = SUMSPL(THETA,Y,D,N,ARCV,+1)                           CHEV   32
              AMC(J)=AMC(J)*2.0/PI                                           CHEV   33
      15      BMC(J) = BMC(J) * 2.0/PI                                       CHEV   34
      C                                                                      CHEV   35
      C       CALCULATIONS COMPLETE-PRINT COEFFICIENTS OUT                   CHEV   36
      C                                                                      CHEV   37
              IF(ISBODY) WRITE (6,22)                                        CHEV   38
              IF(.NOT.ISBODY) WRITE (6,23)                                   CHEV   39
              WRITE(6,24) NAME                                               CHEV   40
              WRITE(6,25) MMAX                                               CHEV   41
              WRITE(6,202) (X(I),I=1,N)                                      CHEV   42
              WRITE(6,203) (AMC(I),I=1,MMAX)                                 CHEV   43
              WRITE(6,204) (BMC(I),I=1,MMAX)                                 CHEV   44
              AREA = AMC(1) * PI / 4.0                                       CHEV   45
              IF (ISBODY) GO TO 800                                          CHEV   46
              WRITE(6,650) AREA                                              CHEV   47
              RETURN                                                         CHEV   48
      800     WRITE(6,850) AREA                                              CHEV   49
              RETURN                                                         CHEV   50
           22 FORMAT (1H1,* CHEBYCHEV COEFFICIENTS OF BODY *)                CHEV   51
           23 FORMAT(1H1,* CHEBYCHEV COEFFICIENTS OF STRUT *)                CHEV   52
           24 FORMAT (1H0,7A10)                                              CHEV   53
      25      FORMAT (1H0,9X,*MMAX = *,I4//)                                 CHEV   54
      200     FORMAT (5E16.8)                                                CHEV   55
      201     FORMAT (7A10)                                                  CHEV   56
      202     FORMAT (//10X,"X ARRAY  "/ (8E16.8))                           CHEV   57
      203     FORMAT (//10X,"  AMC ARRAY  "/  (8E16.8))                      CHEV   58
```

```
  204   FORMAT (//10X,"  BMC ARRAY  "/ (8E16.8))                              CHEV      59
  650   FORMAT ("0AREA COEFFICIENT = AREA/(LSTRUT*TMAX) = ",E16.8)          CHEV      60
  850   FORMAT ("0VOLUME COEFFICIENT = VOLUME/(LBODY*AMAX) = ",E16.8)       CHEV      61
        END                                                                   CHEV      62
```

```
              SUBROUTINE CHECK (AMC,BMC,MMAX,ISBODY)                                CHECK      2
        C                                                                           CHECK      3
        C        THIS SUBROUTINE CHECKS THE DIFFERENCE BETWEEN APPROXIMATED         CHECK      4
        C        OFFSETS AND ORIGINAL OFFSETS                                       CHECK      5
        C                                                                           CHECK      6
              DIMENSION AMC(10),BMC(10)                                             CHECK      7
              COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV            CHECK      8
              COMMON/NAME/INCHAR(8),NAME(7)                                         CHECK      9
              LOGICAL ISBODY                                                        CHECK     10
        C                                                                           CHECK     11
        C     N = NUMBER OF ORIGINAL OFFSETS GIVEN                                  CHECK     12
        C     MMAX = MAX ORDER OF CHEBYCHEV EXPANSION USED                          CHECK     13
              IF (ISBODY) WRITE (6,121)                                             CHECK     14
              IF (.NOT.ISBODY) WRITE (6,122)                                        CHECK     15
          123 WRITE(6,126) MMAX,NAME                                                CHECK     16
              WRITE(6,200)                                                          CHECK     17
              DO 100 I = 1,N                                                        CHECK     18
        C                                                                           CHECK     19
              CALL EVAL(X(I),MMAX,YX,AMC,BMC)                                       CHECK     20
        C                                                                           CHECK     21
              IF (I.EQ.N) GO TO 10                                                  CHECK     22
              XX = (X(I) + X(I+1)) * .5                                             CHECK     23
        C                                                                           CHECK     24
              CALL EVAL (XX,MMAX,YXX,AMC,BMC)                                       CHECK     25
        C                                                                           CHECK     26
              WRITE(6,201)  (Y(I),X(I),YX,XX,YXX)                                   CHECK     27
              GO TO 100                                                             CHECK     28
        10    WRITE(6,201) (Y(I),X(I),YX)                                           CHECK     29
        100   CONTINUE                                                              CHECK     30
              RETURN                                                                CHECK     31
          121 FORMAT(1H1,* CHECK OFFSET OF BODY USING CHEBYCHEV APPROXIMATION*)     CHECK     32
          122 FORMAT(1H1,* CHECK OFFSET OF STRUT USING CHEBYCHEV APPROXIMATION*)    CHECK     33
          126 FORMAT (1H0,*CHECK OUT PUT,MMAX=*,I5,*.FOR*,7A10,/)                   CHECK     34
        200   FORMAT (*0*,5X,*REAL Y*,12X,*X*,12X,*APPROX Y*,27X,*XX*,10X,          CHECK     35
             1                 *APPROX Y(XX)*,/)                                    CHECK     36
        201   FORMAT (1X,3E16.8,16X,2E16.8)                                         CHECK     37
              END                                                                   CHECK     38
```

```
      SUBROUTINE EVAL (XX,MMAX,FFSET,AMC,BMC)                           EVAL      2
C                                                                       EVAL      3
C     THIS PROGRAM EVALUATES THE CHEBYCHEV SERIES AT A GIVEN STATION    EVAL      4
C                                                                       EVAL      5
C     XX = X STATION ON A RANGE OF -1 TO +1                             EVAL      6
C     MMAX = MAX ORDER OF CHEBYCHEV EXPANSION USED                      EVAL      7
C     FFSET = OFFSET FOUND FROM CHEBYCHEV SERIES                        EVAL      8
C                                                                       EVAL      9
      DIMENSION AMC(10), BMC(10)                                        EVAL     10
      FFSET = 0.                                                        EVAL     11
      THETA = ASIN(XX)                                                  EVAL     12
      DO 1 J=1,MMAX                                                     EVAL     13
      AJ = J                                                            EVAL     14
      ARGU = (2.*AJ-1.) * THETA                                         EVAL     15
      ARGV = 2.*AJ * THETA                                              EVAL     16
      UU = COS(ARGU)                                                    EVAL     17
      VV = SIN(ARGV)                                                    EVAL     18
    1 FFSET = FFSET + AMC(J)*UU + BMC(J)*VV                             EVAL     19
      RETURN                                                            EVAL     20
      END                                                               EVAL     21
```

```
      SUBROUTINE SPLINE (X,Y,D,N,BC,KODE,KKODE,ISBODY)                   SPLINE     2
C        THIS SUBROUTINE FITS SMOOTH SPLINE SEGMENTS THROUGH A GIVEN SET SPLINE     3
C           OF DISCRETE DATA POINTS                                      SPLINE     4
      DIMENSION X(N),Y(N),D(N)                                           SPLINE     5
      DIMENSION A(100),B(100),C(100),BC(8)                               SPLINE     6
      LOGICAL ISBODY                                                     SPLINE     7
C                                                                        SPLINE     8
C                                                                        SPLINE     9
C     X ARRAY CONTAINS ABSCISSAS FOR INPUT DATA                          SPLINE    10
C     Y ARRAY CONTAINS ORDINATES FOR INPUT DATA                          SPLINE    11
C     ARRAYS A,B,C ARE MAINLY SUB DIAG., DIAGONAL, AND SUPER DIAG.       SPLINE    12
C     D ARRAY IS THE RIGHT HAND SIDE OF MATRIX EQUATION                  SPLINE    13
C         FINAL ANSWERS ARE PLACED IN D                                  SPLINE    14
C                                                                        SPLINE    15
C     SOLUTION TECHNIQUE IS GAUSSIAN ELIMINATION                         SPLINE    16
C     ARRAY BC CONTAINS SPECIFIED BOUNDARY CONDITIONS                    SPLINE    17
C                                                                        SPLINE    18
C     KODE = 0  USER IS SPECIFYING BOUNDARY CONDITIONS                   SPLINE    19
C     KODE = 1  USER IS TAKING AN EXTRAPOLATION OF SECOND                SPLINE    20
C          DERIVATIVES AS THE BOUNDARY CONDITION                         SPLINE    21
C     KKODE = 0  NO PRINT OUT FROM SPLINE ROUTINE                        SPLINE    22
C     KKODE = 1  SPLINE PRINTS ABSCISSAS,ORDINATES AND SECOND DERIVATIVE SPLINE    23
C                                                                        SPLINE    24
      IF (N.GT.2) GO TO 25                                               SPLINE    25
      D(1) = 0.                                                          SPLINE    26
      D(2) = 0.                                                          SPLINE    27
      NPRINT = 2                                                         SPLINE    28
      GO TO 20                                                           SPLINE    29
   25 IF (N.GT.3) GO TO 50                                               SPLINE    30
      YDD = 2.*((X(3)-X(2))*Y(1)+(X(2)-X(1))*Y(3)-(X(3)-X(1))*Y(2))      SPLINE    31
     1     / ((X(3)-X(2))*(X(2)-X(1))*(X(3)-X(1)))                       SPLINE    32
      D(1) = YDD                                                         SPLINE    33
      D(2) = YDD                                                         SPLINE    34
      D(3) = YDD                                                         SPLINE    35
      NPRINT = 3                                                         SPLINE    36
      GO TO 20                                                           SPLINE    37
   50 NPRINT = 0                                                         SPLINE    38
      II = 1                                                             SPLINE    39
      NL1 = N-1                                                          SPLINE    40
      NL2 = N-2                                                          SPLINE    41
      DO 12 I = 1,N                                                      SPLINE    42
      A(I) = 0.                                                          SPLINE    43
      B(I) = 0.                                                          SPLINE    44
      C(I) = 0.                                                          SPLINE    45
   12 D(I) = 0.                                                          SPLINE    46
C                                                                        SPLINE    47
C        SET UP MATRICES(A TRIDIAGONAL STRUCTURE)                        SPLINE    48
C                                                                        SPLINE    49
      A(1) = (X(3)-X(2))/(X(3)-X(1))                                     SPLINE    50
      C(1) = 2.0                                                         SPLINE    51
      B(1) = 1.0 - A(1)                                                  SPLINE    52
      D(1) = 6.0*((Y(3)-Y(2))/(X(3)-X(2))-(Y(2)-Y(1))/                   SPLINE    53
     1 (X(2)-X(1)))/(X(3)-X(1))                                          SPLINE    54
      H = X(3) - X(2)                                                    SPLINE    55
      DO 19 J=3,NL1                                                      SPLINE    56
      HP = X(J+1) - X(J)                                                 SPLINE    57
      C(J) = HP/(H+HP)                                                   SPLINE    58
```

```
      B(J) = 2.0
      A(J) = 1.0 - C(J)
      D(J) = 6.0*((Y(J+1)-Y(J))/HP-(Y(J)-Y(J-1))/H)/(HP+H)
   19 H = HP
C
C     CHECK BOUNDARY CONDITIONS
C
      IF (KODE .EQ. 1 ) GO TO 6
      IF(BC(2) .EQ. 0.0 .AND. BC(3) .EQ. 0.0) GO TO 1
      GO TO 2
    1 WRITE(6,300) BC(2),BC(3)
      GO TO 6
    2 C(2) = 0.0
      A(2) = BC(3)-BC(2)*(X(2)-X(1))/3.0
      B(2) = -BC(2) * (X(2)-X(1))/6.0
      D(2) = BC(4)-BC(1)*Y(1)-BC(2)*(Y(2)-Y(1))/(X(2)-X(1))
    3 IF(BC(6) .EQ. 0.0 .AND. BC(7) .EQ. 0.0) GO TO 4
      GO TO 5
    4 WRITE(6,301) BC(6),BC(7)
      GO TO 7
    5 C(N) = 0.0
      A(N) = BC(6)*(X(N)-X(N-1))/6.0
      B(N) = BC(6)*(X(N)-X(N-1))/3.0+BC(7)
      D(N) = BC(8)-BC(5)*Y(N)-BC(6)*(Y(N)-Y(N-1))/(X(N)-X(N-1))
      GO TO 10
    6 C(2) = (X(2)-X(1))/(X(3)-X(2))
      A(2) = 1.0
      B(2) = -1.0-C(2)
      D(2) = 0.0
      IF (KODE .EQ. 0) GO TO 3
    7 C(N) = (X(N)-X(N-1))/(X(N-1)-X(N-2))
      A(N) = -1.0 - C(N)
      B(N) = 1.0
      D(N) = 0.0
C
C     SOLVE EQUATIONS
C
   10 C(2) = - A(2)*A(1)/B(1) + C(2)
      DO 13 I=1,NL2
      AUGH = ABS(B(I))
      IF (AUGH .LT. 1.0E-06) GO TO 9
      CONST = A(I+1)/B(I)
      B(I+1) = B(I+1) - CONST*C(I)
      D(I+1) = D(I+1) - CONST*D(I)
      IF (I .NE. NL2) GO TO 13
      A(N) = A(N) - C(N)*C(I) / B(I)
      D(N) = D(N) - C(N)*D(I) / B(I)
      GO TO 13
    9 II = I+1
      D(I) = D(I)/C(I)
      D(I+1) = D(I+1) - B(I+1)*D(I)
      B(I+1) = A(I+1)
      A(I+1) = 0.0
      D(I+2) = D(I+2) - A(I+2)*D(I)
      A(I+2) = 0.0
      IF (I .NE. NL2) GO TO 13
      A(N) = C(N)
```

```
   13   CONTINUE
        DET = B(N-1)*B(N) - C(N-1)*A(N)
        STORE = D(N)
        D(N) = (B(N-1)*D(N) - D(N-1)*A(N))/DET
        D(N-1) = (D(N-1)*B(N) - C(N-1)*STORE)/DET
        IP = 0
        DO 15 I =2,NL2
        JI = N-I
        IF (JI .EQ.IP) GO TO 15
        IF(JI .EQ. II) GO TO 8
        D(JI) = (D(JI)-C(JI)*D(JI+1))/B(JI)
        GO TO 15
    8   IP = JI-1
        STORE = D(JI)
        D(JI) = D(JI-1)
        D(JI-1) = (STORE - C(JI-1)*D(JI+1))/B(JI-1)
   15   CONTINUE
        D(1) = (D(1) - A(1)*D(3) -C(1)*D(2))/B(1)
C
C       ALL DONE
C
   20   IF (KKODE .NE. 1) GO TO 1000
        IF (ISBODY) WRITE(6,100) (I,X(I),Y(I),D(I),I=1,N)
        IF (.NOT.ISBODY) WRITE(6,101) (I,X(I),Y(I),D(I),I=1,N)
        IF (NPRINT.EQ.3) WRITE(6,901)
        IF (NPRINT.EQ.2) WRITE(6,902)
 1000   RETURN
  100   FORMAT (1H1,
     1        3CX,*SPLINE APPROXIMATION OF BODY*,//27X,
     2        *ABSCISSA        ORDINATE           2ND DERIVATIVE*//
     3        (12X,I10,3E16.8))
  101   FORMAT (1H1,30X,*SPLINE APPROXIMATION OF STRUT*,//27X,
     1        *ABSCISSA        ORDINATE           2ND DERIVATIVE*//
     2        (12X,I10,3E16.8))
  300   FORMAT(//"THE BOUNDARY CONDITIONS SPECIFIED AT THE FIRST POINT
     1 ARE NOT SUFFICIENT"/,"  BC(2) = ",E16.8/" BC(3)=",E16.8//" THE
     2 PROGRAM WILL CONTINUE WITH THE SECOND DERIVATIVE AT THE FIRST
     3 POINT EQUAL TO A LINEAR COMBINATION  OF THE SECOND AND THIRD"//)
  301   FORMAT(//"THE BOUNDARY CONDITIONS SPECIFIED AT THE LAST POINT
     1 ARE NOT SUFFICIENT"/," BC(6)=",E16.8/"BC(7)=",E16.8//
     2" THE PROGRAM WILL CONTINUE WITH THE SECOND DERIVATIVE AT THE LAST
     3 POINT EQUAL TO A LINEAR COMBINATION OF THE THIRD AND SECOND
     4 LAST")
  901   FORMAT (*0THREE POINT SPLINE, INTERPOLATION IS QUADRATIC*)
  902   FORMAT (*0TWO POINT SPLINE, INTERPOLATION IS LINEAR*)
        END
```

```
      REAL FUNCTION SUMSPL(X,Y,S,N,AK,KODE)
C
C     THIS PROGRAM USES THE SPLINE COEFFICIENTS AND INTEGRATES THE
C         TRIGONOMETRIC INTEGRALS
C     OF THE FORM F(X)*SIN(AK*X) OR F(X)*COS(AK*X)
C     IF KODE = +1  PROGRAM PERFORMS SINE INTEGRATION
C     IF KODE = -1  PROGRAM PERFORMS COSINE INTEGRATION
C     INPUT DATA ARE GIVEN AS (Y(I) = F(X(I)),I=1,N)
C     N IS THE TOTAL NUMBER OF INPUT POINTS (ABSCISSAS)
C     S-ARRAY STORES THE SECOND DERIVATIVES OF THE SPLINE FUNCTIONS
C     THE RANGE OF INTEGRATION IS FROM X(1) TO X(N)
C
      DIMENSION X(100),Y(100),S(100)
      NL = N-1
      SUMSPL = 0.0
      D2 = AK*AK
      D3 = AK*D2
      D4 = AK*D3
      DO 1 I=1,NL
      H = X(I+1) - X(I)
      IF (H) 1,1,9
    9 ZAA = (S(I+1)-S(I))/(6.*H)
      ZBB = S(I)/2.0
      ZCC = (Y(I+1)-Y(I))/H- H*(S(I+1)+2.0*S(I))/6.
      E = SIN(AK*H)
      F = COS(AK*H)
      AA1 = (3.*D2*H*H-6.)*F/D4+(D2*H*H*H-6.*H)*E/D3+6./D4
      AA2 = 3.*H*H*E/D2-6.*E/D4-H*H*H*F/AK+6.*H*F/D3
      BB1 = 2.*H*F/D2-2.*E/D3+H*H*E/AK
      BB2 = 2.*H*E/D2+2.*F/D3-H*H*F/AK-2./D3
      CC1 = F/D2 + H*E/AK -1.0/D2
      CC2 = E/D2 -H*F/AK
      DD1 = E/AK
      DD2 = -F/AK+1.0/AK
      UU = SIN(AK*X(I))
      VV = COS(AK*X(I))
      IF (KODE)2,3,4
    2 SUMSPL = SUMSPL +VV*(AA1*ZAA+BB1*ZBB+CC1*ZCC+DD1*Y(I))
     1 - UU*(AA2*ZAA+BB2*ZBB+CC2*ZCC+DD2*Y(I))
      GO TO 1
    4 SUMSPL = SUMSPL + UU*(AA1*ZAA+BB1*ZBB+CC1*ZCC+DD1*Y(I)) +
     1 VV*(AA2*ZAA + BB2*ZBB + CC2*ZCC+DD2*Y(I))
    1 CONTINUE
      GO TO 5
    3 WRITE(6,100)
      SUMSPL = 1.0
    5 RETURN
  100 FORMAT (1H1,"KODE = 0"//" SUMSPL IS SET EQUAL TO 1.")
      END
```

```
                SUBROUTINE SPLFIT(X,Y,S,N,T,YVAL,YD,YDD)                        SPLFIT     2
          C                                                                     SPLFIT     3
          C     FIND THE DERIVATIVES (1ST AND 2ND) CF THE SPLINE FITTED DATA    SPLFIT     4
          C                                                                     SPLFIT     5
                DIMENSION X(100),Y(100),S(100)                                  SPLFIT     6
                I = 1                                                           SPLFIT     7
          3     IF (T-X(I)) 6,6,4                                               SPLFIT     8
          4     I = I + 1                                                       SPLFIT     9
                GO TO 3                                                         SPLFIT    10
          6     I = I - 1                                                       SPLFIT    11
                IF (I.EQ.0) I=1                                                 SPLFIT    12
                H = X(I+1) - X(I)                                               SPLFIT    13
                HTX = X(I+1) - T                                                SPLFIT    14
                HTT = T - X(I)                                                  SPLFIT    15
                YVAL = S(I) * (HTX**3)/(6.*H)+S(I+1)*HTT**3/(6.*H)              SPLFIT    16
               1 +(Y(I)-S(I)*H*H/6.)*HTX/H+(Y(I+1)-S(I+1)*H*H/6.)*HTT/H         SPLFIT    17
                YD = -S(I)*HTX*HTX/(2.*H)+S(I+1)*HTT*HTT/(2.*H)                 SPLFIT    18
               1 +(Y(I+1)-Y(I))/H-(S(I+1)-S(I))/6.*H                            SPLFIT    19
                YDD = S(I) *HTX/H+S(I+1)*HTT/H                                  SPLFIT    20
                RETURN                                                          SPLFIT    21
                END                                                             SPLFIT    22
```

```
      FUNCTION FORMDR(VL)                                        FORMDR   2
C                                                                FORMDR   3
C     EVALUATE FORM DRAG COEFFICIENT                             FORMDR   4
C                                                                FORMDR   5
      DIMENSION X(9), Y(9)                                       FORMDR   6
      DATA X / 1.325, 1.38, 1.43, 1.48, 1.55, 1.68, 1.9, 2.2, 2.5 /   FORMDR   7
      DATA Y / .51, .71, .84, .85, .83, .75, .65, .57, .50 /     FORMDR   8
      IF (VL.GE.1.325.AND.VL.LT.2.5) GO TO 4                     FORMDR   9
      FORMDR=0.5                                                 FORMDR  10
      GO TO 5                                                    FORMDR  11
    4 FORMDR = YINTP (VL,X,Y,9)                                  FORMDR  12
    5 RETURN                                                     FORMDR  13
      END                                                        FORMDR  14
```

```
FUNCTION YINTP       74/74    OPT=0 ROUND=*/ TRACE           FTN 4.6+460        04/07/81  13.06.42         PAGE      1

                 FUNCTION YINTP (XA,X,Y,N)                                  YINTP       2
           C                                                                YINTP       3
           C     INTERPOLATING FROM A SET OF DATA (X,Y)                     YINTP       4
           C                                                                YINTP       5
                 DIMENSION X(1), Y(1)                                       YINTP       6
               1 DO 10 I=1,N                                                YINTP       7
                 IF (X(I)-XA) 10,10,2                                       YINTP       8
               2 IN=I-2                                                     YINTP       9
                 IF(IN) 4,4,6                                               YINTP      10
               4 IN=1                                                       YINTP      11
                 GO TO 12                                                   YINTP      12
               6 NN=N-3                                                     YINTP      13
                 IF (IN-NN) 12,12,8                                         YINTP      14
               8 IN = NN                                                    YINTP      15
                 GO TO 12                                                   YINTP      16
              10 CONTINUE                                                   YINTP      17
              12 IO=IN+3                                                    YINTP      18
                 YINTP=0.                                                   YINTP      19
                 DO 20 I=IN,IO                                              YINTP      20
                 PROD=Y(I)                                                  YINTP      21
                 DO 16 J=IN,IO                                              YINTP      22
                 IF (I-J) 15,16,15                                          YINTP      23
              15 PROD = PROD*(XA-X(J))/(X(I)-X(J))                          YINTP      24
              16 CONTINUE                                                   YINTP      25
              20 YINTP=YINTP+PROD                                           YINTP      26
                 RETURN                                                     YINTP      27
                 END                                                        YINTP      28
```

```
            FUNCTION CFITTC (RN)                                       CFITTC      2
      C                                                                CFITTC      3
      C     THIS FUNCTION DETERMINES THE ITTC CORRELATION ALLOWANCE    CFITTC      4
      C                                                                CFITTC      5
            DEN = ALOG10 (RN) - 2.0                                    CFITTC      6
            CFITTC = 0.075 / DEN**2                                    CFITTC      7
            RETURN                                                     CFITTC      8
            END                                                        CFITTC      9
```

PROGRAM LISTING OF SYNTHESIS

```
      PROGRAM SWATH(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8)          SYNTHS   2
C        MAIN PROGRAM FOR THE RESISTANCE AND PROPULSION OF SWATH SHIPS    SYNTHS   3
C             PROGRAMMED BY A. M. REED JUNE 1975                          SYNTHS   4
C                  POWERING MODULE  BY W. G. DAY                          SYNTHS   5
C                  APPEN AGE DRAG MODULE BY J. W. GRANT                   SYNTHS   6
C                      REVISED APRIL 1978 BY E. C. CHAI                   SYNTHS   7
C                                                                         SYNTHS   8
      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX               SYNTHS   9
C                                                                         SYNTHS  10
      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,  SYNTHS  11
     1 CLCB, SEPDIS, CSTRUT, CSTRT2, FDIA, SPAN, CHORD, TFINS, NLOC       SYNTHS  12
C                                                                         SYNTHS  13
      COMMON / OMEGA / VMFPS, GAMAOS, GAMAOB, GOSQ, HSOLS, HBOLB, WETS,   SYNTHS  14
     1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG    SYNTHS  15
C                                                                         SYNTHS  16
      COMMON / PHYSCO / RHO,GNU,G,PI,DELCF                                SYNTHS  17
C                                                                         SYNTHS  18
      COMMON/PSI/NPTSZ,PTSAF,EXPN,NALMAX,NAL,TAIL,ALFA                    SYNTHS  19
     1      ,ALSMAX,NSTEPS                                                SYNTHS  20
C                                                                         SYNTHS  21
      DIMENSION TITLE(8), VCFF(24), EHP(24), SHP(24)                      SYNTHS  22
C                                                                         SYNTHS  23
C        DEFINE THE PSI CONSTANTS                                         SYNTHS  24
C                                                                         SYNTHS  25
C                                                                         SYNTHS  26
C                                                                         SYNTHS  27
C                                                                         SYNTHS  28
C        READ AND CHECK INPUT                                             SYNTHS  29
C                                                                         SYNTHS  30
       CALL RIN(TITLE)                                                    SYNTHS  31
C                                                                         SYNTHS  32
C        CALCULATE  THE OMEGA AND BODY CONSTANTS                          SYNTHS  33
C                                                                         SYNTHS  34
      HSOLS  = HS/XLS                                                     SYNTHS  35
      HBOLB  = HB/XLB                                                     SYNTHS  36
      RATIOL =XLB/XLS                                                     SYNTHS  37
      B = CSTRUT/XLS                                                      SYNTHS  38
      D = 0.0                                                             SYNTHS  39
      IF ( NLOC .NE. 0 ) D = CSTRT2/XLS                                   SYNTHS  40
      SFINS = SPAN*CHORD                                                  SYNTHS  41
      TBFINS = 0.01*TFINS                                                 SYNTHS  42
      IF(XLB .GT. 30.0 .AND. TBFINS .LT. 0.041666) TBFINS = 0.041666      SYNTHS  43
      IF(XLB .LE. 30.0 .AND. TBFINS .LT. 0.005208) TBFINS = 0.005208      SYNTHS  44
      SBFINS = TBFINS*SPAN                                                SYNTHS  45
      SQRTL = SQRT(XLS)                                                   SYNTHS  46
      BDIA = SQRT(AX/PI)                                                  SYNTHS  47
C                                                                         SYNTHS  48
C        DETERMINE THE CHEBYSHEV COEFICIENTS FOR THE SHIP                 SYNTHS  49
C             CALCULATE THE THE WETTED SURFACE OF THE BODY AND STRUT      SYNTHS  50
C             PLOT THE STRUT PROFILE AND SECTIONAL AREA CURVE             SYNTHS  51
C                                                                         SYNTHS  52
      CALL CHEB(TITLE)                                                    SYNTHS  53
C                                                                         SYNTHS  54
C        DETERMINE THE SPEEDS AND PROCEED THROUGH THE RESISTANCE LOOP     SYNTHS  55
C                                                                         SYNTHS  56
      READ(5,100) VKMIN, VKMAX, NC                                        SYNTHS  57
      IF ( NC .GT. 20 ) GO TO 90                                          SYNTHS  58
```

```
      DV = (VKMAX-VKMIN)*1.6878/FLOAT(NC-1)                             SYNTHS    59
      VMIN = VKMIN*1.6878                                               SYNTHS    60
      NV = NC + 2                                                       SYNTHS    61
      DO 20 I=1, NV                                                     SYNTHS    62
         VOFF(I) = VMIN + DV*FLOAT(I-1)                                 SYNTHS    63
         VMFPS = VOFF(I)                                                SYNTHS    64
C                                                                       SYNTHS    65
C           DETERMINE THE FRICTIONAL DRAG COEFFICIENTS                  SYNTHS    66
C                                                                       SYNTHS    67
         RNS = XLS*VMFPS/GNU                                            SYNTHS    68
         CFS= CFITTC(RNS)                                               SYNTHS    69
         RNB = XLB*VMFPS/GNU                                            SYNTHS    70
         CFB = CFITTC(RNB)                                              SYNTHS    71
C                                                                       SYNTHS    72
C           DETERMINE THE WAVE RESISTANCE MATRIX                        SYNTHS    73
C                                                                       SYNTHS    74
         GAMAOS = G*XLS*.5 / VMFPS**2                                   SYNTHS    75
         GOSQ = GAMAOS**2                                               SYNTHS    76
         GAMAOB = GAMAOS*RATIOL                                         SYNTHS    77
         PHIS = 2.0*HSOLS/GAMAOS                                        SYNTHS    78
         PHIB = 2.0*HBOLB/GAMAOB                                        SYNTHS    79
         CALL RWAVE(B, D, NLOC)                                         SYNTHS    80
C                                                                       SYNTHS    81
C           DETERMINE THE APPENDAGE DRAG                                SYNTHS    82
C                                                                       SYNTHS    83
         IF ( SPAN .EQ. 0.0 )GO TO 16                                   SYNTHS    84
            CALL FINDRG(RHO, GNU, VMFPS, SFINS, TFINS, CHORD, TBFINS,   SYNTHS    85
     1         SBFINS, APPDRG)                                          SYNTHS    86
         GO TO 18                                                       SYNTHS    87
   16    CONTINUE                                                       SYNTHS    88
            APPDRG = 0.0                                                SYNTHS    89
   18    CONTINUE                                                       SYNTHS    90
C                                                                       SYNTHS    91
C           PRINT DRAG OUTPUT                                           SYNTHS    92
C                                                                       SYNTHS    93
         CALL ROUT(TITLE, EHP(I))                                       SYNTHS    94
   20 CONTINUE                                                          SYNTHS    95
C                                                                       SYNTHS    96
C        DETERMINE A PROPELLER DESIGN AND CALCULATE THE SHP             SYNTHS    97
C                                                                       SYNTHS    98
      VDES = VOFF(NV-2)                                                 SYNTHS    99
      EHPDES = EHP(NV-2)                                                SYNTHS   100
      CALL SHPCMP(VDES,EHPDES,PDIA,BDIA,HB,VOFF,EHP,SHP,NV,TITLE)       SYNTHS   101
C                                                                       SYNTHS   102
C        PRINT THE POWERING OUTPUT                                      SYNTHS   103
C                                                                       SYNTHS   104
      WRITE(6,101) (TITLE(I), I=1, 8)                                   SYNTHS   105
      DO 40 I = 1, NV                                                   SYNTHS   106
         VK = VOFF(I)/1.6878                                            SYNTHS   107
         SLR = VK/SQRTL                                                 SYNTHS   108
         PC = EHP(I)/SHP(I)                                             SYNTHS   109
         WRITE(6,102) VK, VOFF(I), SLR, EHP(I), SHP(I), PC              SYNTHS   110
   40 CONTINUE                                                          SYNTHS   111
      STOP                                                              SYNTHS   112
   90 CONTINUE                                                          SYNTHS   113
      WRITE(6,103) NC                                                   SYNTHS   114
      STOP                                                              SYNTHS   115
```

```
        100   FORMAT(2F10.5,I5)                                                SYNTHS   116
        101   FORMAT(1H1,8A10//T5,"V(KNOTS)",T16,"V(FT/SEC)",T27,"SLR(STRUT)",  SYNTHS   117
            1 T46,"EHP",T58,"SHP",T71,"PC")                                    SYNTHS   118
        102   FORMAT(6(2X,F10.3))                                              SYNTHS   119
        103   FORMAT("-****ERROR NC GREATER THAN 20, NC = ",I3)                SYNTHS   120
              END                                                              SYNTHS   121
```

```
              BLOCK DATA                                                  BLDATA    2
C                                                                         BLDATA    3
              COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX        BLDATA    4
C                                                                         BLDATA    5
              COMMON / PHYSCO / RHO,GNU,G,PI,DELCF                        BLDATA    6
C                                                                         BLDATA    7
              COMMON/PSI/NPTSZ,PTSAF,EXPN,NALMAX,NAL,TAIL,ALFA            BLDATA    8
             1     ,ALSMAX,NSTEPS                                         BLDATA    9
C                                                                         BLDATA   10
              COMMON/PLOT/NFIRST,NLAST,NPOINT,XMAX,XMIN,NSCL1,NCHAR,NSCALE(4),  BLDATA   11
             1          PCHAR(2)                                          BLDATA   12
              COMMON/XRPLOTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H  BLDATA   13
C                                                                         BLDATA   14
              LOGICAL NSCL1                                               BLDATA   15
C                                                                         BLDATA   16
C                MAXIMUM ORDER OF CHEBYCHEV APPROXIMATION                 BLDATA   17
C                                                                         BLDATA   18
              DATA MMAX / 3 /                                             BLDATA   19
C                                                                         BLDATA   20
C                DEFINE PHYSICAL CONSTANTS                                BLDATA   21
C                                                                         BLDATA   22
              DATA RHO / 1.9908 /                                         BLDATA   23
              DATA GNU / 1.297E-5 /                                       BLDATA   24
              DATA G / 32.174 /                                           BLDATA   25
              DATA PI / 3.1415926535898 /                                 BLDATA   26
              DATA DELCF / 0.0005 /                                       BLDATA   27
C                                                                         BLDATA   28
C                DEFINE CONSTANTS FOR NUMERICAL INTEGRATION               BLDATA   29
C                                                                         BLDATA   30
              DATA NPTSZ / 31 /                                           BLDATA   31
              DATA PTSAF / 10.0 /                                         BLDATA   32
              DATA EXPN / 7.0 /                                           BLDATA   33
              DATA NALMAX / 300 /                                         BLDATA   34
              DATA ALSMAX / 26.0 /                                        BLDATA   35
C                                                                         BLDATA   36
C                 DEFINE DATA FOR PLOT ROUTINE                            BLDATA   37
C                                                                         BLDATA   38
              DATA NHL, NSBH, NVL, NSBV, HCHAR, VCHAR, PCHAR(1), PCHAR(2) /  BLDATA   39
             1      6,   10,   11,   10,   1H.,    1H.,    1HS,    1HB     /  BLDATA   40
C                                                                         BLDATA   41
              DATA NFIRST, NLAST, NPOINT / 1, 101, 1 /                    BLDATA   42
C                                                                         BLDATA   43
              DATA XMAX, XMIN, NSCL1 / 1.0, -1.0, .TRUE. /                BLDATA   44
C                                                                         BLDATA   45
              DATA NCHAR / 0 /                                            BLDATA   46
C                                                                         BLDATA   47
              DATA NSCALE /0, 3, 0, 3 /                                   BLDATA   48
C                                                                         BLDATA   49
              END                                                         BLDATA   50
```

```
      SUBROUTINE CHEB(TITLE)
C
      COMMON / COEFS , AS  ^), BSM(3), ABM(3), BBM(3), MMAX
C
      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,
     1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC
C
      COMMON / OMEGA / VMFPS, GAMAOS, GAMAOB, GOSQ, HSOLS, HBOLB, WETS,
     1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG
C
      COMMON / PHYSCO / RHO,GNU,G,PI,DELCF
C
      DIMENSION TITLE(8)
C
C        DETERMINE THE CHEBYSHEV COEFFICIENTS
C
      ASM(1) = 4.0*CWP/PI
      ASM(2) = (1.0 - 16.0*CIYY)*ASM(1)
      ASM(3) = 1.0 - ASM(1) - ASM(2)
      BSM(1) = 4.0*CLCF*ASM(1)
      BSM(2) = 0.0
      BSM(3) = 0.0
      ABM(1) = 4.0*CP/PI
      ABM(2) = 1.0 - ABM(1)
      ABM(3) = 0.0
      BBM(1) = 4.0*CLCB*ABM(1)
      BBM(2) = 0.0
      BBM(3) = 0.0
C
C        DETERMINE THE WETTED SURFACE
C
      CALL WSURFB(WETB)
      CALL WSURFS(WETS)
      WETB = WETB - XLS*TSMAX*CWP
      IF ( NLOC .NE. 0 ) WETB = WETB - XLS*TSMAX*CWP
      WETFIN = 2.0*CHORD*SPAN
      WTSURF = WETB + WETS + WETFIN
      IF ( NLOC .NE. 0 ) WTSURF = WTSURF + WETS
C
C        PRINT THE CHEBYSHEV COEFFICIENTS AND THE WETTED SURFACE
C
      WRITE(6,340) (TITLE(I),I = 1, 8 )
      WRITE(6,330) (ASM(M),M=1,MMAX)
      WRITE(6,335) (BSM(M),M=1,MMAX)
      WRITE(6,345) (ABM(K),K=1,MMAX)
      WRITE(6,350) (BBM(K),K=1,MMAX)
      WRITE(6,357) WETS,WETB, WETFIN, WTSURF
C
C        PLOT THE STRUT WATER LINE AND THE BODY AREA DISTRIBUTION
C
      CALL PCHEB (ASM,BSM,ABM,BBM,MMAX,TITLE)
      RETURN
  330 FORMAT(1H0,33X,*STRUT CHEBYCHEV COEFFICIENTS ASM(M)*/(10X,5E16.8))
  335 FORMAT(1H0,33X,*STRUT CHEBYCHEV COEFFICIENTS BSM(M)*/(10X,5E16.8))
  340 FORMAT("1",8A10)
  345 FORMAT(1H0,33X,*BODY CHEBYCHEV COEFFICIENTS ABM(M)*/(10X,5E16.8))
  350 FORMAT(1H0,33X,*BODY CHEBYCHEV COEFFICIENTS BBM(M)*/(10X,5E16.8))
```

```
  357  FORMAT(1H0,4X,* STRUT WETTED SURFACE IN SQUARE FEET  =*,E14.7,     CHEB      59
      1/          4X,* BODY WETTED SURFACE IN SQUARE FEET  =*,E14.7,      CHEB      60
      2/          4X,* FOIL WETTED SURFACE IN SQUARE FEET  =*,E14.7,      CHEB      61
      3/          4X,* TOTAL WETTED SURFACE IN SQUARE FEET  =*,E14.7)     CHEB      62
       END                                                                CHEB      63
```

```
      SUBROUTINE RIN(TITLE)                                             RIN    2
C                                                                       RIN    3
      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP, RIN    4
     1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC     RIN    5
C                                                                       RIN    6
      DIMENSION TITLE(8)                                                RIN    7
C                                                                       RIN    8
C     INPUT                                                             RIN    9
C                                                                       RIN   10
C        TITLE  = LABEL PRINTED ON OUTPUT                               RIN   11
C        XLS    = LENGTH OF STRUT (FT)                                  RIN   12
C        HS     = DRAFT OF STRUT (FT)                                   RIN   13
C        TSMAX  = MAXIMUM THICKNESS OF STRUT (FT)                       RIN   14
C        CWP    = WATERPLANE AREA COEFICIENT                            RIN   15
C            CWP = AWP / (TSMAX * LS )                                  RIN   16
C            AWP = LS * (TSMAX/2) * (PI/2) * AS1                        RIN   17
C        CLCF   = WATERPLANE MOMENT COEFFICIENT                         RIN   18
C            CLCF = MY(STRUT) / (AWP*LS)                                RIN   19
C            MY(STRUT) = (TSMAX,2) * (LS/2)**2 * (PI/2) * BS1           RIN   20
C        CIYY   = WATERPLANE INERTIA COEFFICIENT                        RIN   21
C            CIYY = IYY / (AWP*LS**2)                                   RIN   22
C            IYY = (1/2) * (TSMAX/2) * (LS/2)**3 * (PI/2) * ( A1 - A2 ) RIN   23
C        XLB    = LENGTH OF BODY (FT)                                   RIN   24
C        HB     = DEPTH OF SUBMERGENCE TO BODY CENTERLINE (FT)          RIN   25
C        AX     = MAXIMUM BODY CROSS-SECTIONAL AREA (FT**2)             RIN   26
C        CP     = BODY PRISMATIC COEFFICIENT                            RIN   27
C            CP = DISPLACEMENT(BODY) / (AX *LB )                        RIN   28
C            DISPLACEMENT(BODY) = LB * AX * (PI/4) * AB1                RIN   29
C        CLCB   = BODY MOMENT COEFFICIENT                               RIN   30
C            CLCB = MY(BODY) / (AX*LB**2)                               RIN   31
C            MY(BODY) = AX * (LB/2)**2 * (PI/4) * BB1                   RIN   32
C        SEPDIS = SEPARATION DISTANCE OF HULL CENTERLINES (FT)          RIN   33
C        CSTRUT = DISTANCE OF CL OF STRUT FROM CL OF BODY (FT)          RIN   34
C           ( + FORWARD OF THE BODY CL, - AFT OF THE THE BODY CL )      RIN   35
C        CSTRT2 = DISTANCE OF CL OF SECOND STRUT ( IF PRESENT ) FROM    RIN   36
C           CL OF BODY (FT)                                             RIN   37
C        NLOC   = SWITCH INDICATING THE PRESENCE OF A SECOND STRUT      RIN   38
C            NLOC = 0 - ONE STRUT ONLY                                  RIN   39
C            NLOC = 1 - TWO STRUTS                                      RIN   40
C        PDIA   = PROPELLER DIAMETER (FT)                               RIN   41
C        SPAN   = SPAN OF A DEMI-FIN (FT)                               RIN   42
C        CHORD  = CHORD OF A DEMI-FIN (FT)                              RIN   43
C        TFINS  = MAXIMUM THICKNESS OF A DEMI-FIN (FT)                  RIN   44
C                                                                       RIN   45
C     READ INPUT                                                        RIN   46
C                                                                       RIN   47
      READ (5,210) (TITLE(K),K=1,8)                                     RIN   48
      READ(5,215) XLS,HS,TSMAX                                          RIN   49
      READ (5,215) CWP, CLCF, CIYY                                      RIN   50
      READ(5,215) XLB,HB,AX                                             RIN   51
      READ(5,215) CP, CLCB                                              RIN   52
      READ(5,215) SEPDIS                                                RIN   53
      READ(5,216) CSTRUT, CSTRT2, NLOC                                  RIN   54
      READ(5,215) PDIA                                                  RIN   55
      READ(5,215) SPAN,CHORD,TFINS                                      RIN   56
C                                                                       RIN   57
C        ECHO INPUT                                                     RIN   58
```

```
      C                                                              RIN    59
            WRITE(6,310) (TITLE(K),K=1,8)                            RIN    60
            WRITE(6,325) XLS,HS,TSMAX                                RIN    61
            WRITE(6,330) CWP, CLCF, CIYY                             RIN    62
            WRITE(6,340) XLB,HB,AX                                   RIN    63
            WRITE(6,335) CP, CLCB                                    RIN    64
            WRITE(6,355) SEPDIS                                      RIN    65
            IF ( NLOC .NE. 0 ) GO TO 20                              RIN    66
               WRITE(6,356) CSTRUT                                   RIN    67
            GO TO 25                                                 RIN    68
      20    CONTINUE                                                 RIN    69
               WRITE(6,357) CSTRUT, CSTRT2                           RIN    70
      25    CONTINUE                                                 RIN    71
            WRITE(6,358) PDIA                                        RIN    72
            WRITE(6,359) SPAN, CHORD, TFINS                          RIN    73
            RETURN                                                   RIN    74
      C                                                              RIN    75
      C        FORMATS                                               RIN    76
      C                                                              RIN    77
      210   FORMAT(8A10)                                             RIN    78
      215   FORMAT(8F10.5)                                           RIN    79
      216   FORMAT(2F10.5,I5)                                        RIN    80
      310   FORMAT(1H1,///,3X,* INPUT DATA FOR *,8A10//)             RIN    81
      325   FORMAT(1H ,3X,35H STRUT GEOMETRIC CHARACTERISTICS   .//,5X,   RIN    82
           1      8H  XLS  =,F12.6,9X,8H   HS  =,F12.6,9X,8H TSMAX =,F12.6)   RIN    83
      330   FORMAT(" WATERPLANE AREA COEFFICIENT  = ",F12.6/         RIN    84
           1" WATERPLANE MOMENT COEFICIENT  = ",F12.6/               RIN    85
           2" WATERPLANE INERTIA COEFICIENT  =" F12.6)               RIN    86
      335   FORMAT(" BODY PRISMATIC COEFFICIENT  = ",F10.5/          RIN    87
           1 " BODY MOMENT COEFFICIENT  = ",F10.5)                   RIN    88
      340   FORMAT(1H0,3X,35H BODY GEOMETRIC CHARACTERISTICS    .//,5X,   RIN    89
           1      8H  XLB  =,F12.6,9X,8H   HB  =,F12.6,9X,8H   AX  =,F12.6)   RIN    90
      355   FORMAT(1H0,5X,* SEPARATION DISTANCE IN FEET    =*,8F8.3)   RIN    91
      356   FORMAT(*0      STRUT LOCATION IN FEET    =*,8(2X,F10.4))  RIN    92
      357   FORMAT(" FORWARD STRUT LOCATION IN FEET  = ",F10.4/      RIN    93
           1 " AFT STRUT LOCATION IN FEET  = ",F10.4)                RIN    94
      358   FORMAT(" PROPELLER DIAMETER IN FEET  = ",F10.4)          RIN    95
      359   FORMAT(" FIN SPAN IN FEET  = ",F10.4/                    RIN    96
           1 " FIN CHORD IN FEET  = ",F10.5/                         RIN    97
           2 " FIN THICKNESS IN FEET  = ",F10.4)                     RIN    98
            END                                                      RIN    99
```

```
      SUBROUTINE ROUT(TITLE, EHP)                                       ROUT   2
C                                                                       ROUT   3
      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP, ROUT   4
     1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC     ROUT   5
C                                                                       ROUT   6
      COMMON / OMEGA / VMFPS, GAMAOS, GAMAOB, GOSQ, HSOLS, HBOLB, WETS, ROUT   7
     1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG   ROUT   8
C                                                                       ROUT   9
      COMMON / PHYSCO / RHO,GNU,G,PI,DELCF                              ROUT  10
C                                                                       ROUT  11
      DIMENSION TITLE(8)                                                ROUT  12
C                                                                       ROUT  13
C        SUM = A(TRANSPOSE)*T*A + B(TRANSPOSE)*W*B                      ROUT  14
C                                                                       ROUT  15
      CALL SUM(SUM1S,SUM1B,SUM1SB,SUM12S,SUM12B,SM12SB)                 ROUT  16
C                                                                       ROUT  17
C        COMPUTE THE CONSTANTS NEEDED FOR OUTPUT                        ROUT  18
C                                                                       ROUT  19
      FROUDS = VMFPS/SQRT(32.155*XLS)                                   ROUT  20
      VLS = FROUDS*SQRT(G)/1.6878                                       ROUT  21
      FROUDB = VMFPS/SQRT(32.155*XLB)                                   ROUT  22
      VLB = FROUDB*SQRT(G)/1.6878                                       ROUT  23
      VMKNTS = VMFPS/1.6878                                             ROUT  24
      B2OLB = SEPDIS/XLB                                                ROUT  25
      CSOLB=CSTRUT/XLB                                                  ROUT  26
      IF ( NLOC .NE. 0 ) CSOLB2 = CSTRT2/XLB                            ROUT  27
      AAS = (TSMAX/HS)*(PI/2.0)*GAMAOS                                  ROUT  28
      AAB = 2.0*PI*(AX/XLB**2)*(GAMAOB/GAMAOS**2)                       ROUT  29
      AASB = (2.0*PI)*AX/(HS*XLS)                                       ROUT  30
      AAWTSF = (0.5*RHO*WTSURF*VMFPS**2)/1000.                          ROUT  31
C                                                                       ROUT  32
C        COMPUTE THE WAVE RESISTANCE COEFFICIENTS                       ROUT  33
C                                                                       ROUT  34
      CMS1 = AAS*SUM1S                                                  ROUT  35
      R1S = CMS1*RHO*G*TSMAX*HS*XLS                                     ROUT  36
      CWS1    = R1S/AAWTSF                                              ROUT  37
C                                                                       ROUT  38
      CMB1 = AAB*SUM1B                                                  ROUT  39
      R1B = CMB1*RHO*G*AX*XLB                                           ROUT  40
      CWB1    = R1B/AAWTSF                                              ROUT  41
C                                                                       ROUT  42
      CMSB1 = AASB*SUM1SB                                               ROUT  43
      R1SB = CMSB1*RHO*G*TSMAX*HS*XLS                                   ROUT  44
      CWSB1   = R1SB/AAWTSF                                             ROUT  45
C                                                                       ROUT  46
      CMS12 = AAS*SUM12S                                                ROUT  47
      R12S = CMS12*RHO*G*TSMAX*HS*XLS                                   ROUT  48
      CWS12   = R12S/AAWTSF                                             ROUT  49
C                                                                       ROUT  50
      CMB12 = AAB*SUM12B                                                ROUT  51
      R12B = CMB12*RHO*G*AX*XLB                                         ROUT  52
      CWB12   = R12B/AAWTSF                                             ROUT  53
C                                                                       ROUT  54
      CMSB12 = AASB*SM12SB                                              ROUT  55
      R12SB = CMSB12*RHO*G*TSMAX*HS*XLS                                 ROUT  56
      CWSB12 = R12SB/AAWTSF                                             ROUT  57
C                                                                       ROUT  58
```

```
      CW1 = CWS1+CWB1+CWSB1                                             ROUT  59
      CW12 = CWS12+CWB12+CWSB12                                         ROUT  60
      CW = CW1+CW12                                                     ROUT  61
      RW=CW*AAWTSF                                                      ROUT  62
C                                                                       ROUT  63
C        COMPUTE RESIDUAL RESISTANCE                                    ROUT  64
C                                                                       ROUT  65
      CFORM=FORMDR(VLS)                                                 ROUT  66
      CR = CFORM + CW                                                   ROUT  67
      RR = CR * AAWTSF                                                  ROUT  68
C                                                                       ROUT  69
C        COMPUTE THE FRICTIONAL DRAG COEFFICIENTS                       ROUT  70
C                                                                       ROUT  71
      RFS=0.5*RHO*VMFPS*VMFPS*WETS*(CFS+DELCF)                          ROUT  72
      RFB=0.5*RHO*VMFPS*VMFPS*WETB*(CFB+DELCF)                          ROUT  73
      RF=RFS+RFB                                                        ROUT  74
      CFSP=RFS/AAWTSF                                                   ROUT  75
      CFBP=RFB/AAWTSF                                                   ROUT  76
      CF=CFSP+CFBP                                                      ROUT  77
      CAPP = APPDRG/AAWTSF                                              ROUT  78
C                                                                       ROUT  79
C        TOTAL DRAG                                                     ROUT  80
C                                                                       ROUT  81
      RT=RF+RR+APPDRG                                                   ROUT  82
      CT=CF+CR+CAPP                                                     ROUT  83
      EHP = RT*VMFPS/550.0                                              ROUT  84
C                                                                       ROUT  85
C        OUTPUT                                                         ROUT  86
C                                                                       ROUT  87
      WRITE(6,410) (TITLE(K),K=1,8)                                     ROUT  88
      WRITE(6,325) XLS,HS,TSMAX                                         ROUT  89
      WRITE(6,415)       GAMAOS,FROUDS,HSOLS                            ROUT  90
      WRITE(6,340) XLB,HB,AX                                            ROUT  91
      WRITE(6,420)       GAMAOB,FROUDB,HBOLB                            ROUT  92
      WRITE(6,356) SEPDIS,B2OLB                                         ROUT  93
      IF ( NLOC .NE. 0 ) GO TO 20                                       ROUT  94
         WRITE(6,358) CSTRUT,CSOLB                                      ROUT  95
      GO TO 25                                                          ROUT  96
   20 CONTINUE                                                          ROUT  97
         WRITE(6,359) CSTRUT, CSOLB, CSTRT2, CSOLB2                     ROUT  98
   25 CONTINUE                                                          ROUT  99
      WRITE (6,360) VMFPS,VMKNTS,VLS,VLB                                ROUT 100
      WRITE(6,354)                                                      ROUT 101
      WRITE(6,430) R1S,CWS1                                             ROUT 102
      WRITE(6,440) R1B,CWB1                                             ROUT 103
      WRITE(6,445) R1SB,CWSB1,CW1                                       ROUT 104
      WRITE(6,450) R12S,CWS12                                           ROUT 105
      WRITE(6,455) R12B,CWB12                                           ROUT 106
      WRITE(6,460) R12SB,CWSB12,CW12,RW,CW                              ROUT 107
      WRITE(6,461) CFORM,CR,RR                                          ROUT 108
      WRITE(6,465) RFS, CFSP, RFB, CFBP, RF, CF                         ROUT 109
      WRITE(6,470) APPDRG, CAPP                                         ROUT 110
      WRITE(6,475) RT, CT, EHP                                          ROUT 111
      RETURN                                                            ROUT 112
C                                                                       ROUT 113
C         FORMATS                                                       ROUT 114
C                                                                       ROUT 115
```

```
  325  FORMAT(1H ,3X,35H STRUT GEOMETRIC CHARACTERISTICS   ,//,5X,
      1     8H  XLS  =,F12.6,9X,8H   HS  =,F12.6,9X,8H TSMAX =,F12.6)
  340  FORMAT(1H0,3X,35H BODY GEOMETRIC CHARACTERISTICS    ,//,5X,
      1     8H  XLB  =,F12.6,9X,8H   HB  =,F12.6,9X,8H   AX  =,F12.6)
  354  FORMAT(//T25,"RESISTANCES IN POUNDS ARE FOR A DEMI-HULL")
  356  FORMAT(*0      SEPARATION DISTANCE IN FEET     =*,F8.3,
      1     5X,*  B/LB              =*,F7.3)
  358  FORMAT(*0      STRUT CL FROM BODY CL IN FEET   =*,F8.3,
      1    5X,*STRUT OFSET/LB          =*,F7.3)
  359  FORMAT(*0    FORWARD  STRUT CL FROM BODY CL IN FEET   =*,F8.3,
      1    5X,*FORWARD STRUT OFSET/LB           =*,F7.3//
      2          *     AFT  STRUT CL FROM BODY CL IN FEET   =*,F8.3,
      3    5X,*AFT STRUT OFSET/LB             =*,F7.3)
  360  FORMAT(1H0,5X,*  SHIP SPEED IS *,F8.3,* FPS, *,F8.3,* KNOTS, *,
      1     6X,*  STRUT V-L RATIO =*,F8.3,6X,*BODY V-L RATIO = *,F8.3)
  410  FORMAT(*1     *,     35H WAVE RESISTANCE CALCULATIONS FOR  ,8A10)
  415  FORMAT(1H ,4X,                        8HGAMAOS =,F12.6,9X,8HFROUDS =,
      1     F12.6,9X,8H HS/LS =,F12.6)
  420  FORMAT(1H ,4X,                        8HGAMAOB =,F12.6,9X,8HFROUDB =,
      1     F12.6,9X,8H HB/LB =,F12.6)
  430  FORMAT(/T30,32H STRUT WAVE DRAG IN POUNDS   =  ,E14.7,/30X,
      2     32H R1S/(RHO/2*WTSURF*V**2)      =  ,F10.3,4HE-03)
  440  FORMAT(/T30,32H BODY WAVE DRAG IN POUNDS    =  ,E14.7,/30X,
      2     32H R1B/(RHO/2*WTSURF*V**2)      =  ,F10.3,4HE-03)
  445  FORMAT(/T30,32H STRUT-BODY WAVE DRAG IN LBS =  ,E14.7,/30X,
      2     32H R1SB/(RHO/2*WTSURF*V**2)     =  ,F10.3,4HE-03,
      3     10X,10H CW1 TOT =,F10.3,4HE-03)
  450  FORMAT(/T30,32H STRUT INTERFERENCE DRAG LBS =  ,E14.7,/30X,
      2     32H R12S/(RHO/2*WTSURF*V**2)     =  ,F10.3,4HE-03)
  455  FORMAT(/T30,32H BODY INTERFERENCE DRAG  LBS = ,E14.7,/30X,
      2     32H R12B/(RHO/2*WTSURF*V**2)     =  ,F10.3,4HE-03)
  460  FORMAT(/T30,32H STRUT-BODY INTFRNCE DRAG LB =  ,E14.7,/30X,
      2     32H R12SB/(RHO/2*WTSURF*V**2)    =  ,F10.3,4HE-03,
      3     10X,10H CW2 TOT =,F10.3,4HE-03,//,
      4 86X," RW TOTAL = ",E14.7,/
      5 86X," CW TOTAL = ",F10.3,"E-03")
   461 FORMAT(/30X,*       CFORM              = *,F10.3,4HE-03,
      1/30X,*RESIDUAL DRAG COEFF. =  *,F10.3,*E-03*,
      2 /30X,*RESIDUAL RESISTANCE IN LBS= *,E14.7)
  465  FORMAT(30X,32H STRUT FRICTION DRAG IN LBS  =  ,E14.7,
      1 /30X,32H RFS/(RHO/2*WTSURF*V**2)      =  ,F10.3,4HE-03,
      2 /30X,32H BODY FRICTION DRAG IN LBS    =  ,E14.7,
      3 /30X,32H RFB/(RHO/2*WTSURF*V**2)      =  ,F10.3,4HE-03,
      4 /86X," RF TOTAL = ",E14.7,
      5 /86X," CF TOTAL = ",F10.3,"E-03")
  470  FORMAT(/30X,"APPENDAGE DRAG IN POUNDS      =  ",E14.7,
      1 /30X,"APPENDAGE DRAG COEFFICIENT   =  ",F10.3,"E-03")
  475  FORMAT(//86X," RT        = ",E14.7,
      1 /86X," CT        = ",F10.3,"E-03",
      2 /86X," EHP       = ",F10.3)
       END
```

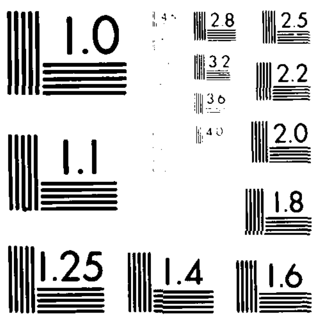
```
      SUBROUTINE RWAVE(B,D,NLOC2)                                       RWAVE      2
C                                                                       RWAVE      3
                                                                        RWAVE      4
C                                                                       RWAVE      5
C        SUBROUTINE FOR COMPUTING AUXILIARY FUNCTIONS T AND W           RWAVE      6
C             USING CHEBYCHEV EXPANSION - PROGRAM VERSION OF 8/05/72    RWAVE      7
C             MODIFICATIONS BY A M REED OCTOBER 1974                    RWAVE      8
C                                                                       RWAVE      9
C                                                                       RWAVE     10
      COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),      RWAVE     11
     1 WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3), RWAVE     12
     2 WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)       RWAVE     13
C                                                                       RWAVE     14
      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX             RWAVE     15
C                                                                       RWAVE     16
      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP, RWAVE     17
     1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC     RWAVE     18
C                                                                       RWAVE     19
      COMMON / OMEGA / VMFPS, GAMAOS, GAMAOB, GOSQ, HSOLS, HBOLB, WETS,  RWAVE     20
     1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG   RWAVE     21
C                                                                       RWAVE     22
C                                                                       RWAVE     23
      COMMON / PHYSCO / RHO,GNU,G,PI,DELCF                              RWAVE     24
C                                                                       RWAVE     25
      COMMON/PSI/NPTSZ,PTSAF,EXPN,NALMAX,NAL,TAIL,ALFA                  RWAVE     26
     1      ,ALSMAX,NSTEPS                                              RWAVE     27
C                                                                       RWAVE     28
      DIMENSION SIMP(100),VJALFA(7)                                     RWAVE     29
C                                                                       RWAVE     30
      MMAX2=MMAX*2                                                      RWAVE     31
C                                                                       RWAVE     32
C        SET THE T AND W ARRAYS TO ZERO                                 RWAVE     33
C                                                                       RWAVE     34
      CALL RINIT                                                        RWAVE     35
C                                                                       RWAVE     36
C        INTEGRATION OF T AND W FROM ALPHA EQUALS GAMMAOS TO GAMMAOS+1.0 RWAVE     37
C             MAKE TRIGNOMETRIC SUBSTITUTION TO AVOID SQUAREROOT        RWAVE     38
C             SINGULARITY AT ALPHA EQUALS GAMMAOS                       RWAVE     39
C                                                                       RWAVE     40
C        ZETA IS THE VARIABLE OF INTEGRATION--ZETA=SQRT(ALPHA-GAMMAOS)  RWAVE     41
C        NPTSZ IS THE NUMBER OF  POINTS TO TAKE IN THE INTERVAL         RWAVE     42
C             0.0 LESS THAN OR EQUAL ZETA LESS THAN OR EQUAL 1.0        RWAVE     43
C        AINCR IS THE STEP SIZE FOR THE ZETA INTEGRATION                RWAVE     44
C        ALFAZ IS THE VALUE OF ALPHA FOR ZETA VARYING FROM ZERO TO ONE  RWAVE     45
C                                                                       RWAVE     46
      SEP = (2.0/GAMAOS)*SEPDIS/XLS                                     RWAVE     47
      CALL SIMPSN(NPTSZ,SIMP)                                           RWAVE     48
      XNPTZ=FLOAT(NPTSZ)                                                RWAVE     49
      AINCR = 1.0/(XNPTZ-1.0)                                           RWAVE     50
      HZETA = AINCR/3.0                                                 RWAVE     51
      DO 30 I=1,NPTSZ                                                   RWAVE     52
         WTINT = SIMP(I) * HZETA                                        RWAVE     53
         EI=FLOAT(I-1)                                                  RWAVE     54
         ZETA = AINCR*EI                                                RWAVE     55
         ALFAZ = GAMAOS+ZETA**2                                         RWAVE     56
         ALFASQ=ALFAZ**2                                                RWAVE     57
         SQZTA=2.0/SQRT(GAMAOS+ALFAZ)                                   RWAVE     58
```

```
      SEPCOS = COS(SEP*ALFAZ*SQRT(ALFASQ-GOSQ))
C
      CALL RINTEG(ALFAZ,B,D,NLOC2,WTINT,SEPCOS,SQZTA)
C
30    CONTINUE
C
C        INTEGRATION FOR T AND W FROM ALPHA EQUALS GAMMAOS+1.0 TO ALMAX
C
      NAL =  0
      ALFA = GAMAOS + 1.
      PWT = 0.
      ALMAX = SQRT((2.302585*EXPN*GAMAOS) / (2.*HBOLB*RATIOL)
     1              + (GAMAOS+1.)**2 )
210   CONTINUE
      ISIMP = 1
      IF (ALFA.GT.ALMAX) GO TO 215
      IF (NAL.LT.NALMAX) GO TO 212
      WRITE (6,302)
      GO TO 215
212   CONTINUE
      NAL = NAL + 2
      DQDA = (2.0*ALFASQ-GOSQ)/SQRT(ALFASQ-GOSQ)
      SSIZE=(2.*PI)/((2.*RATIOL+DQDA*SEP)*PTSAF)
      TWT= SSIZE/3.
      WTINT = TWT + PWT
      PWT = TWT
      GO TO 250
215   CONTINUE
      ISIMP=3
      WTINT = PWT
      GO TO 250
220   CONTINUE
      WTINT = 4.*TWT
      ISIMP = 2
250   CONTINUE
      ALFASQ=ALFA**2
      SQ=1.0/SQRT(ALFASQ-GOSQ)
      SEPCOS = COS(SEP*ALFA/SQ)
C
      CALL RINTEG(ALFA,B,D,NLOC2,WTINT,SEPCOS,SQ)
C
      ALFA = ALFA + SSIZE
      GO TO (220,210,270), ISIMP
C
C        ADD TAIL INTEGRATION TO TS AND WS
C
270   CONTINUE
      ALFA = ALFA - SSIZE
      NSTEPS = 0
      IF (ALSMAX.LE.ALFA) GO TO 278
      SSIZE = PI/PTSAF
      ANSTEP = (ALSMAX-ALFA)/SSIZE
      NSTEPS = IFIX(ANSTEP/2.)*2 + 2
      SSIZE = (ALSMAX-ALFA) / NSTEPS
      WT = SSIZE/3.
      ANWT = 1.
      NDO = NSTEPS + 1
```

AD-A099 533 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 20/4
DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTI--ETC(U)
APR 81 A M REED
UNCLASSIFIED DTNSRDC/SPD-0927-02 NL

3 OF 3

END
DATE
FILMED
6-81
DTIC



MICROCOPY RESOLUTION TEST CHART

```
      CON=1.0                                                           RWAVE    116
      DO 277 IDO = 1,NDO                                                RWAVE    117
         ALFASQ=ALFA**2                                                 RWAVE    118
         IF(NLOC2.NE.0) CON=2.0*(1.0+COS((B-D)*ALFA*2.0))               RWAVE    119
      CALL BESSJ (ALFA,MMAX2,VJALFA)                                    RWAVE    120
         VS = PHIS*ALFASQ                                               RWAVE    121
         SQ=1.0/SQRT(ALFASQ-GOSQ)                                       RWAVE    122
         ESA = 1. - EXP (-VS)                                           RWAVE    123
         ES = SQ * ESA**2 / ALFASQ                                      RWAVE    124
         DO 276 M = 1,MMAX                                              RWAVE    125
            MAA = 2*M                                                   RWAVE    126
            MBB = MAA + 1                                               RWAVE    127
            DO 275 N = M, MMAX                                          RWAVE    128
               NAA = 2*N                                                RWAVE    129
               NBB = NAA + 1                                            RWAVE    130
               WTINT = ANWT * WT                                        RWAVE    131
               TSA = WTINT*ES*VJALFA(MAA)*VJALFA(NAA)*CON               RWAVE    132
               WSA = WTINT*ES*VJALFA(MBB)*VJALFA(NBB)*CON               RWAVE    133
               TS(M,N) = TS(M,N) + TSA                                  RWAVE    134
               WS(M,N) = WS(M,N) + WSA                                  RWAVE    135
  275       CONTINUE                                                    RWAVE    136
  276    CONTINUE                                                       RWAVE    137
         ALFA = ALFA + SSIZE                                            RWAVE    138
         BNWT = 2.                                                      RWAVE    139
         IF (ANWT.LE.2.) BNWT = 4.                                      RWAVE    140
         IF (IDO.EQ.NDO-1) BNWT = 1.                                    RWAVE    141
         ANWT = BNWT                                                    RWAVE    142
  277 CONTINUE                                                          RWAVE    143
  278 CONTINUE                                                          RWAVE    144
      ALFASQ = ALFA**2                                                  RWAVE    145
      RAT = GAMAOS/ALFA                                                 RWAVE    146
      IF (RAT.LT.1.E-4) GO TO 280                                       RWAVE    147
      TAIL=(-SQRT(ALFASQ-GOSQ)/ALFASQ+ASIN(RAT)/GAMAOS)/(2.0*PI*GOSQ)   RWAVE    148
      GO TO 290                                                         RWAVE    149
  280 CONTINUE                                                          RWAVE    150
      TAIL=1.0/(3.0*PI*ALFA*ALFASQ)                                     RWAVE    151
C                                                                       RWAVE    152
C        CALCULATE TAIL RESULTS                                         RWAVE    153
C                                                                       RWAVE    154
  290 CONTINUE                                                          RWAVE    155
      DO 296 M = 1,MMAX                                                 RWAVE    156
         DO 295 N = M, MMAX                                             RWAVE    157
            SIGN = (-1)**(M+N)                                          RWAVE    158
            TS(M,N) = TS(M,N) - SIGN*TAIL                               RWAVE    159
            WS(M,N) = WS(M,N) + SIGN*TAIL                               RWAVE    160
  295    CONTINUE                                                       RWAVE    161
  296 CONTINUE                                                          RWAVE    162
C                                                                       RWAVE    163
C        MULTIPLY BY CONSTANTS AND REFLECT SYMMETRIC MATRICIES)         RWAVE    164
C                                                                       RWAVE    165
      CALL REFLKT                                                       RWAVE    166
C                                                                       RWAVE    167
      RETURN                                                            RWAVE    168
  302 FORMAT (*1ALFA INTEGRATION REACHED NALMAX*)                       RWAVE    169
      END                                                               RWAVE    170
```

```
      SUBROUTINE RINTEG(ALFA,B,D,NLOC2,WTINT,SEPCOS,SQ)               RINTEG     2
C                                                                     RINTEG     3
C        NUMERICAL INTEGRATION FOR AUXILIARY FUNCTIONS T AND W        RINTEG     4
C                                                                     RINTEG     5
      COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),    RINTEG     6
     1 WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3), RINTEG  7
     2 WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)     RINTEG     8
C                                                                     RINTEG     9
      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX           RINTEG    10
C                                                                     RINTEG    11
      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP, RINTEG  12
     1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC   RINTEG    13
C                                                                     RINTEG    14
      COMMON / OMEGA / VMFPS, GAMAOS, GAMAOB, GOSQ, HSOLS, HBOLB, WETS, RINTEG  15
     1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG RINTEG   16
C                                                                     RINTEG    17
      DIMENSION VJALFA(7),VJBETA(7)                                   RINTEG    18
C                                                                     RINTEG    19
C        MMAX IS MAXIMUM NUMBER OF TERMS IN CHEBCHEV SERIES           RINTEG    20
C                                                                     RINTEG    21
      MMAX2 = 2*MMAX                                                  RINTEG    22
      BETA = RATIOL*ALFA                                              RINTEG    23
      ALFASQ=ALFA**2                                                  RINTEG    24
C                                                                     RINTEG    25
C        CALCULATE FACTORS FOR STRUT NOT CENTERED                     RINTEG    26
C                                                                     RINTEG    27
      CO=COS(B*ALFA*2.0)                                              RINTEG    28
      SI=SIN(B*ALFA*2.0)                                              RINTEG    29
      CON=1.0                                                         RINTEG    30
      IF (NLOC2.EQ.0) GO TO 32                                        RINTEG    31
C                                                                     RINTEG    32
C        CALCULATE FACTORS FOR TANDEM STRUTS                          RINTEG    33
C                                                                     RINTEG    34
      CO=(CO+COS(D*ALFA*2.0))                                         RINTEG    35
      SI=(SI+SIN(D*ALFA*2.0))                                         RINTEG    36
      CON=2.0*(1.0+COS((B-D)*ALFA*2.0))                               RINTEG    37
   32 CONTINUE                                                        RINTEG    38
C                                                                     RINTEG    39
C        COMPUTE AUXILIARY FUNCTIONS VJALFA AND VJBETA                RINTEG    40
C                                                                     RINTEG    41
      CALL BESSJ (ALFA, MMAX2, VJALFA )                               RINTEG    42
      CALL BESSJ ( BETA, MMAX2, VJBETA )                              RINTEG    43
C                                                                     RINTEG    44
C        COMPUTE AUXILIARY FUNCTIONS ES AND EB FOR ALPHA FROM         RINTEG    45
C                                                                     RINTEG    46
      VS=PHIS*ALFASQ                                                  RINTEG    47
      VB = PHIB*BETA**2                                               RINTEG    48
      ESA = 1.0                                                       RINTEG    49
      EBA = 0.0                                                       RINTEG    50
      IF(VS .LT. 300.) ESA = ESA-EXP(-VS)                             RINTEG    51
      IF(VB .LT. 300.) EBA = EXP(-VB)                                 RINTEG    52
      ES=SQ*ESA**2/ALFASQ                                             RINTEG    53
      EB=SQ*EBA**2*ALFASQ                                             RINTEG    54
      ESB = SQ*ESA*EBA                                                RINTEG    55
C                                                                     RINTEG    56
C        TB,TS,WB,WS,TW12,TS12,WB12, AND WS12 ARE ALL SYMMETRIC WRT M RINTEG    57
C             AND N SO WE NEED ONLY CALCULATE THE UPPER HALF OF THESE RINTEG    58
```

```
      C           MATRICIES                                                 RINTEG    59
      C                                                                     RINTEG    60
            DO 56 M=1,MMAX                                                  RINTEG    61
               MAA = 2*M                                                    RINTEG    62
               MBB = 2*M+1                                                  RINTEG    63
               DO 55 N=M,MMAX                                               RINTEG    64
                  NAA = 2*N                                                 RINTEG    65
                  NBB = 2*N+1                                               RINTEG    66
                  TSA =   WTINT*ES*VJALFA(MAA)*VJALFA(NAA)*CON              RINTEG    67
                  WSA =   WTINT*ES*VJALFA(MBB)*VJALFA(NBB)*CON              RINTEG    68
                  IF(EB .EQ. 0.0) GO TO 54                                  RINTEG    69
                  TBA =   WTINT*EB*VJBETA(MAA)*VJBETA(NAA)                  RINTEG    70
                  WBA =   WTINT*EB*VJBETA(MBB)*VJBETA(NBB)                  RINTEG    71
                  TSBA =   WTINT*ESB*VJBETA(NAA)*VJALFA(MAA)*CO             RINTEG    72
                  TSBAP= - WTINT*ESB*VJBETA(NAA)*VJALFA(MBB)*SI             RINTEG    73
                  WSBA =   WTINT*ESB*VJBETA(NBB)*VJALFA(MBB)*CO             RINTEG    74
                  WSBAP=   WTINT*ESB*VJBETA(NBB)*VJALFA(MAA)*SI             RINTEG    75
                  TSBB =   WTINT*ESB*VJBETA(MAA)*VJALFA(NAA)*CO             RINTEG    76
                  TSBBP= - WTINT*ESB*VJBETA(MAA)*VJALFA(NBB)*SI             RINTEG    77
                  WSBB =   WTINT*ESB*VJBETA(MBB)*VJALFA(NBB)*CO             RINTEG    78
                  WSBBP=   WTINT*ESB*VJBETA(MBB)*VJALFA(NAA)*SI             RINTEG    79
                  TB(M,N) = TB(M,N)+TBA                                     RINTEG    80
                  WB(M,N) = WB(M,N)+WBA                                     RINTEG    81
                  TSB(M,N) = TSB(M,N)+TSBA                                  RINTEG    82
                  TSBP(M,N) = TSBP(M,N)+TSBAP                               RINTEG    83
                  WSB(M,N) = WSB(M,N)+WSBA                                  RINTEG    84
                  WSBP(M,N) = WSBP(M,N)+WSBAP                               RINTEG    85
                  IF(M.EQ.N) GO TO 54                                       RINTEG    86
                  TSB(N,M) = TSB(N,M)+TSBB                                  RINTEG    87
                  TSBP(N,M) = TSBP(N,M)+TSBBP                               RINTEG    88
                  WSB(N,M) = WSB(N,M)+WSBB                                  RINTEG    89
                  WSBP(N,M) = WSBP(N,M)+WSBBP                               RINTEG    90
         54       CONTINUE                                                  RINTEG    91
                  TS(M,N) = TS(M,N)+TSA                                     RINTEG    92
                  WS(M,N) = WS(M,N)+WSA                                     RINTEG    93
                  TS12(M,N) = TS12(M,N)+TSA*SEPCOS                          RINTEG    94
                  WS12(M,N) = WS12(M,N)+WSA*SEPCOS                          RINTEG    95
                  IF(EBA .EQ. 0.0) GO TO 55                                 RINTEG    96
                  TB12(M,N) = TB12(M,N)+TBA*SEPCOS                          RINTEG    97
                  WB12(M,N) = WB12(M,N)+WBA*SEPCOS                          RINTEG    98
                  TSB12(M,N) = TSB12(M,N)+TSBA*SEPCOS                       RINTEG    99
                  TSB12P(M,N) = TSB12P(M,N)+TSBAP*SEPCOS                    RINTEG   100
                  WSB12(M,N) = WSB12(M,N)+WSBA*SEPCOS                       RINTEG   101
                  WSB12P(M,N) = WSB12P(M,N)+WSBAP*SEPCOS                    RINTEG   102
                  IF(M.EQ.N) GO TO 55                                       RINTEG   103
                  TSB12(N,M) = TSB12(N,M)+TSBB*SEPCOS                       RINTEG   104
                  TSB12P(N,M) = TSB12P(N,M)+TSBBP*SEPCOS                    RINTEG   105
                  WSB12(N,M) = WSB12(N,M)+WSBB*SEPCOS                       RINTEG   106
                  WSB12P(N,M) = WSB12P(N,M)+WSBBP*SEPCOS                    RINTEG   107
         55    CONTINUE                                                     RINTEG   108
         56 CONTINUE                                                        RINTEG   109
            RETURN                                                          RINTEG   110
            END                                                             RINTEG   111
```

```
      SUBROUTINE RINIT                                                   RINIT     2
C                                                                        RINIT     3
C        INITIALIZE T AND W ARRAY                                        RINIT     4
C                                                                        RINIT     5
      COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),       RINIT     6
     1 WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3), RINIT     7
     2 WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)        RINIT     8
C                                                                        RINIT     9
      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX              RINIT    10
C                                                                        RINIT    11
      DO 5 M=1,MMAX                                                      RINIT    12
         DO 4 N=M,MMAX                                                   RINIT    13
            TS(M,N)  = 0.0                                               RINIT    14
            WS(M,N)  = 0.0                                               RINIT    15
            TB(M,N)  = 0.0                                               RINIT    16
            WB(M,N)  = 0.0                                               RINIT    17
            TSB(M,N) = 0.0                                               RINIT    18
            TSBP(M,N) = 0.0                                              RINIT    19
            TSB(N,M)=0.0                                                 RINIT    20
            TSBP(N,M)=0.0                                                RINIT    21
            WSB(M,N) = 0.0                                               RINIT    22
            WSBP(M,N) = 0.0                                              RINIT    23
            WSB(N,M)=0.0                                                 RINIT    24
            WSBP(N,M)=0.0                                                RINIT    25
            TS12(M,N) = 0.0                                              RINIT    26
            WS12(M,N) = 0.0                                              RINIT    27
            TB12(M,N) = 0.0                                              RINIT    28
            WB12(M,N) = 0.0                                              RINIT    29
            TSB12(M,N) = 0.0                                             RINIT    30
            TSB12P(M,N) = 0.0                                            RINIT    31
            TSB12(N,M)=0.0                                               RINIT    32
            TSB12P(N,M)=0.0                                              RINIT    33
            WSB12(M,N) = 0.0                                             RINIT    34
            WSB12P(M,N) = 0.0                                            RINIT    35
            WSB12(N,M)=0.0                                               RINIT    36
            WSB12P(N,M)=0.0                                              RINIT    37
    4    CONTINUE                                                        RINIT    38
    5 CONTINUE                                                           RINIT    39
      RETURN                                                             RINIT    40
      END                                                                RINIT    41
```

```
              SUBROUTINE REFLKT                                                  REFLKT     2
        C                                                                        REFLKT     3
        C        REFLECT THE SYMMETRICAL MATRICES OF T AND W                     REFLKT     4
        C                                                                        REFLKT     5
              COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),       REFLKT     6
             1 WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3), REFLKT     7
             2 WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)        REFLKT     8
        C                                                                        REFLKT     9
              COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX              REFLKT    10
        C                                                                        REFLKT    11
              DO 64 M = 1, MMAX                                                  REFLKT    12
                 EM=FLOAT(M)                                                     REFLKT    13
                 U = 2.0*EM-1.0                                                  REFLKT    14
                 DO 63 N=M,MMAX                                                  REFLKT    15
                    EN=FLOAT(N)                                                  REFLKT    16
                    FOURMN=4.0*EM*EN                                             REFLKT    17
                    UV=U*(2.0*EN-1.0)                                            REFLKT    18
                    TS(M,N) = TS(M,N)*UV                                         REFLKT    19
                    TS(N,M) = TS(M,N)                                            REFLKT    20
                    WS(M,N) = WS(M,N)*FOURMN                                     REFLKT    21
                    WS(N,M) = WS(M,N)                                            REFLKT    22
                    TB(M,N) = TB(M,N)*UV                                         REFLKT    23
                    TB(N,M) = TB(M,N)                                            REFLKT    24
                    WB(M,N) = WB(M,N)*FOURMN                                     REFLKT    25
                    WB(N,M) = WB(M,N)                                            REFLKT    26
                    TSB(M,N) = TSB(M,N)*UV                                       REFLKT    27
                    TSBP(M,N) = TSBP(M,N)*UV                                     REFLKT    28
                    WSB(M,N) = WSB(M,N)*FOURMN                                   REFLKT    29
                    WSBP(M,N) = WSBP(M,N)*FOURMN                                 REFLKT    30
                    IF (M.EQ.N) GO TO 62                                         REFLKT    31
                    TSB(N,M)=TSB(N,M)*UV                                         REFLKT    32
                    TSBP(N,M)=TSBP(N,M)*UV                                       REFLKT    33
                    WSB(N,M)=WSB(N,M)*FOURMN                                     REFLKT    34
                    WSBP(N,M)=WSBP(N,M)*FOURMN                                   REFLKT    35
         62         CONTINUE                                                     REFLKT    36
                    TS12(M,N) = TS12(M,N)*UV                                     REFLKT    37
                    TS12(N,M) = TS12(M,N)                                        REFLKT    38
                    WS12(M,N) = WS12(M,N)*FOURMN                                 REFLKT    39
                    WS12(N,M) = WS12(M,N)                                        REFLKT    40
                    TB12(M,N) = TB12(M,N)*UV                                     REFLKT    41
                    TB12(N,M) = TB12(M,N)                                        REFLKT    42
                    WB12(M,N) = WB12(M,N)*FOURMN                                 REFLKT    43
                    WB12(N,M) = WB12(M,N)                                        REFLKT    44
                    TSB12(M,N) = TSB12(M,N)*UV                                   REFLKT    45
                    TSB12P(M,N) = TSB12P(M,N)*UV                                 REFLKT    46
                    WSB12(M,N) = WSB12(M,N)*FOURMN                               REFLKT    47
                    WSB12P(M,N) = WSB12P(M,N)*FOURMN                             REFLKT    48
                    IF(M.EQ.N) GO TO 63                                          REFLKT    49
                    TSB12(N,M) = TSB12(N,M)*UV                                   REFLKT    50
                    TSB12P(N,M) = TSB12P(N,M)*UV                                 REFLKT    51
                    WSB12(N,M) = WSB12(N,M)*FOURMN                               REFLKT    52
                    WSB12P(N,M) = WSB12P(N,M)*FOURMN                             REFLKT    53
         63      CONTINUE                                                        REFLKT    54
         64   CONTINUE                                                           REFLKT    55
              RETURN                                                             REFLKT    56
              END                                                                REFLKT    57
```

```
      FUNCTION CFITTC(RN)                                              CFITTC     2
C                                                                      CFITTC     3
C         THIS FUNCTION DETERMINES THE ITTC CORRELATION ALLOWANCE      CFITTC     4
C                                                                      CFITTC     5
                                                                       CFITTC     6
C         THIS FUNCTION DETERMINES THE ITTC CORRELATION ALLOWANCE      CFITTC     7
C                                                                      CFITTC     8
      DEN=ALOG10(RN)-2.0                                               CFITTC     9
      CFITTC=0.075/DEN**2                                              CFITTC    10
      RETURN                                                           CFITTC    11
      END                                                              CFITTC    12
```

```
      FUNCTION EVAL(X,A,B,MAX)                                     EVAL   2
C                                                                  EVAL   3
C        THIS FUNCTION EVALUATES THE CHEBYSHEV SERIES              EVAL   4
C             F(X)=SUMMATION M=1,MAX  A(M)*U(M,X)+B(M)*V(M,X)      EVAL   5
C                                                                  EVAL   6
      DIMENSION A(MAX),B(MAX)                                      EVAL   7
      THETA=ASIN(X)                                                EVAL   8
      T=0.0                                                        EVAL   9
      DO 10 M=1,MAX                                                EVAL  10
         XM=FLOAT(M)                                               EVAL  11
         ARGU=(2.0*XM-1.0)*THETA                                   EVAL  12
         ARGV=2.0*XM*THETA                                         EVAL  13
         U=COS(ARGU)                                               EVAL  14
         V=SIN(ARGV)                                               EVAL  15
         T=T+A(M)*U+B(M)*V                                         EVAL  16
   10 CONTINUE                                                     EVAL  17
      EVAL=T                                                       EVAL  18
      RETURN                                                       EVAL  19
      END                                                          EVAL  20
```

```
            SUBROUTINE WSURFB(AREA)                                          WSURFB     2
      C                                                                      WSURFB     3
      C        THIS SUBROUTINE DETERMINES THE SURFACE AREA OF A BODY OF      WSURFB     4
      C             REVOLUTION GIVEN BY CHEBYSHEV COEFFICIENTS               WSURFB     5
      C                                                                      WSURFB     6
            COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX            WSURFB     7
      C                                                                      WSURFB     8
            COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP, WSURFB   9
           1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC    WSURFB    10
      C                                                                      WSURFB    11
            COMMON / PHYSCO / RHO,GNU,G,PI,DELCF                             WSURFB    12
      C                                                                      WSURFB    13
            RAD=SQRT(AX/PI)                                                  WSURFB    14
            CON=(2.0*RAD/XLB)**2                                             WSURFB    15
            DX=0.05                                                          WSURFB    16
            DXSQ=DX*DX                                                       WSURFB    17
            R=0.0                                                            WSURFB    18
            AREA=0.0                                                         WSURFB    19
            DO 10 I=1,40                                                     WSURFB    20
               RL=R                                                          WSURFB    21
               X=DX*FLOAT(I)-1.0                                             WSURFB    22
               R=EVAL(X,ABM,BBM,MMAX)                                        WSURFB    23
            IF (R.LT.0.0) R=0.0                                              WSURFB    24
               R=SQRT(R)                                                     WSURFB    25
               RM=(RL+R)*0.5                                                 WSURFB    26
               DR=R-RL                                                       WSURFB    27
               AREA=AREA+RM*SQRT(DXSQ+CON*DR**2)                             WSURFB    28
         10 CONTINUE                                                         WSURFB    29
            AREA=PI*RAD*XLB*AREA                                             WSURFB    30
            RETURN                                                           WSURFB    31
            END                                                              WSURFB    32
```

```
      SUBROUTINE WSURFS(AREA)                                            WSURFS     2
C                                                                        WSURFS     3
C        THIS SUBROUTINE DETERMINES THE SURFACE AREA OF A STRUT WHOSE    WSURFS     4
C              THICKNESS DISTRIBUTION IS GIVEN BY CHEBYSHEV COEFFICIENTS WSURFS     5
C                                                                        WSURFS     6
      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX              WSURFS     7
C                                                                        WSURFS     8
      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP, WSURFS     9
     1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC      WSURFS    10
C                                                                        WSURFS    11
      CON=(TSMAX/XLS)**2                                                 WSURFS    12
      DX=0.05                                                            WSURFS    13
      DXSQ=DX**2                                                         WSURFS    14
      AREA=0.0                                                           WSURFS    15
      T=0.0                                                              WSURFS    16
      DO 10 I=1,40                                                       WSURFS    17
         TL=T                                                            WSURFS    18
         X=DX*FLOAT(I)-1.0                                               WSURFS    19
         T=EVAL(X,ASM,BSM,MMAX)                                          WSURFS    20
         DT=T-TL                                                         WSURFS    21
         AREA=AREA+SQRT(DXSQ+CON*DT**2)                                  WSURFS    22
   10 CONTINUE                                                           WSURFS    23
      AREA=XLS*HS*AREA                                                   WSURFS    24
      RETURN                                                             WSURFS    25
      END                                                                WSURFS    26
```

```
      SUBROUTINE SIMPSN(NPTS,SIMP)                                      SIMPSN   2
C                                                                       SIMPSN   3
C     SET UP SIMPSON"S MULTIPLIERS FOR INTEGRATION                      SIMPSN   4
C     ODD NUMBER OF POINTS GREATER THAN 1 REQUIRED                      SIMPSN   5
C                                                                       SIMPSN   6
      DIMENSION SIMP(NPTS)                                              SIMPSN   7
      IF((NPTS/2)*2.NE.NPTS) GO TO 3                                    SIMPSN   8
      WRITE(6,90)                                                       SIMPSN   9
      NPTS =NPTS+1                                                      SIMPSN  10
    3 CONTINUE                                                          SIMPSN  11
      SIMP(1) = 1.0                                                     SIMPSN  12
      SIMP(NPTS) = 1.0                                                  SIMPSN  13
      NPTSM1=NPTS-1                                                     SIMPSN  14
      SM=2.0                                                            SIMPSN  15
      DO 5 J=2,NPTSM1                                                   SIMPSN  16
         SM=6.0-SM                                                      SIMPSN  17
         SIMP(J)=SM                                                     SIMPSN  18
    5 CONTINUE                                                          SIMPSN  19
      RETURN                                                            SIMPSN  20
   90 FORMAT(1H0,*EVEN NO. OF POINTS GIVEN - ADDITIONAL POINT SUPPLIED*) SIMPSN  21
      END                                                               SIMPSN  22
```

```
      SUBROUTINE SUM(S 'M1S,SUM1B,SUM1SB,SUM12S,SUM12B,SM12SB)
C
C        SUM COMPUTES THE MATRIX PRODUCTS
C             SUM = A(TRANSPOSE) * T * A + B(TRANSPOSE) * W * B
C
      COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),
     1 WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3),
     2 WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)
C
      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
C
      SUMA1 = 0.0
      SUMB1 = 0.0
      SUMA2 = 0.0
      SUMB2 = 0.0
      SUMA3 = 0.0
      SUMB3 = 0.0
      SUMA4 = 0.0
      SUMB4 = 0.0
      SUMA5 = 0.0
      SUMB5 = 0.0
      SUMA6 = 0.0
      SUMB6 = 0.0
      DO 20 M=1,MMAX
         DO 10 N=1,MMAX
            SUMA1  = SUMA1+ASM(M)*ASM(N)*TS(M,N)
            SUMB1  = SUMB1+BSM(M)*BSM(N)*WS(M,N)
            SUMA2  = SUMA2+ABM(M)*ABM(N)*TB(M,N)
            SUMB2  = SUMB2+BBM(M)*BBM(N)*WB(M,N)
            SUMA3  = SUMA3+ABM(N)*(ASM(M)*TSB(M,N)+BSM(M)*TSBP(M,N))
            SUMB3  = SUMB3+BBM(N)*(BSM(M)*WSB(M,N)+ASM(M)*WSBP(M,N))
            SUMA4  = SUMA4+ASM(M)*ASM(N)*TS12(M,N)
            SUMB4  = SUMB4+BSM(M)*BSM(N)*WS12(M,N)
            SUMA5  = SUMA5+ABM(M)*ABM(N)*TB12(M,N)
            SUMB5  = SUMB5+BBM(M)*BBM(N)*WB12(M,N)
            SUMA6  = SUMA6+ABM(N)*(ASM(M)*TSB12(M,N)+BSM(M)*TSB12P(M,N))
            SUMB6  = SUMB6+BBM(N)*(BSM(M)*WSB12(M,N)+ASM(M)*WSB12P(M,N))
   10    CONTINUE
   20 CONTINUE
      SUM1S = SUMA1+SUMB1
      SUM1B = SUMA2+SUMB2
      SUM1SB = SUMA3+SUMB3
      SUM12S = SUMA4+SUMB4
      SUM12B = SUMA5+SUMB5
      SM12SB = SUMA6+SUMB6
      RETURN
      END
```

```
      SUBROUTINE FINDRG(RHOS,XNUS,VSFW,SFINS,TFINS,CFINS,TBFINS,         FINDRG   2
     +SBFINS,DS)                                                          FINDRG   3
C                                                                         FINDRG   4
C     ************************************************************        FINDRG   5
C       THESE THREE ROUTINES (FINDRG, FOILSC, AND FRICT) ARE EXCERPTS     FINDRG   6
C       FROM CODING DOCUMENTED IN 'AN INVESTIGATION OF APPENDAGE DRAG"    FINDRG   7
C       BY MARC P. LASKY   (NSRDC SPD EVALUATION REPORT 458-H-01)         FINDRG   8
C       THIS PACKAGE WAS COMPLETED ON 25 JUN 1975 BY JERRY GRANT          FINDRG   9
C        NSRDC CODE 1524 THE WORK WAS FUNDED UNDER 1-1507-100-60          FINDRG  10
C     ************************************************************        FINDRG  11
C                                                                         FINDRG  12
C     CLACULATION OF THICKNESS-CHORD RATIOS AND FULL-SCALE PARAMETERS     FINDRG  13
C                                                                         FINDRG  14
      TOC = TFINS / CFINS                                                 FINDRG  15
      TOC2=TOC*TOC                                                        FINDRG  16
      TOC4=TOC2*TOC2                                                      FINDRG  17
      TFINS2 = TFINS * TFINS                                              FINDRG  18
      IF(TBFINS.LE.0) GO TO 3105                                          FINDRG  19
      CLOTB = CFINS / TBFINS                                              FINDRG  20
 3105 CONTINUE                                                            FINDRG  21
C                                                                         FINDRG  22
C        FLAT PLATE FRICTION COEFFICIENT BASED ON LOCAL REYNOLDS NUMBER   FINDRG  23
C                                                                         FINDRG  24
      CALL FRICT(CFSFIN,VSFW,CFINS,XNUS)                                  FINDRG  25
C                                                                         FINDRG  26
C        CONSTANTS TO CONVERT DRAG COEFFICIENTS TO RESISTANCE IN POUNDS   FINDRG  27
C           XS = SHIP CONSTANT                                            FINDRG  28
C                                                                         FINDRG  29
      XS=.5*RHOS*SFINS*VSFW*VSFW                                          FINDRG  30
C                                                                         FINDRG  31
C        CALCULATION OF DRAG COMPONENTS FOR A SYMMETRICAL FOIL SECTION    FINDRG  32
C                                                                         FINDRG  33
      CALL FOILSC(XS,CFSFIN,TOC,TOC2,TOC4,SFINS,TFINS2,SBFINS,CLOTB,RFS,  FINDRG  34
     +RFVAS,RPS,RINTS,RBS)                                                FINDRG  35
C                                                                         FINDRG  36
C        TOTAL OF COMPONENT DRAGS                                         FINDRG  37
C                                                                         FINDRG  38
      DS=RFS+RFVAS+RPS+RINTS+RBS                                          FINDRG  39
C                                                                         FINDRG  40
C        COMPONENTS OF FIN DRAG ARE DEFINED IN SUBROUTINE  FOILSC         FINDRG  41
C           ASSUMPTION - - - NO INDUCED DRAG DUE TO LIFT - - -            FINDRG  42
C                                                                         FINDRG  43
      RETURN                                                              FINDRG  44
      END                                                                 FINDRG  45
```

```
          SUBROUTINE FOILSC(XS,CFS,TOC,TOC2,TOC4,SS,TS2,SBS,CLOTB,RFS,        FOILSC      2
         +RFVAS,RPS,RINTS,RBS)                                                FOILSC      3
      C                                                                       FOILSC      4
      C      FLAT FRICTIONAL RESISTANCE                                       FOILSC      5
      C                                                                       FOILSC      6
          RFS=XS*2.*CFS                                                       FOILSC      7
      C                                                                       FOILSC      8
      C      RESISTANCE DUE TO VELOCITY AUGMENTATION                          FOILSC      9
      C                                                                       FOILSC     10
          RFVAS=XS*4.*CFS*TOC                                                 FOILSC     11
      C                                                                       FOILSC     12
      C      RESISTANCE DUE TO PRESSURE OR SEPARATION (VISCOUS IN NATURE)     FOILSC     13
      C                                                                       FOILSC     14
          RPS=XS*120.*CFS*TOC4                                                FOILSC     15
      C                                                                       FOILSC     16
      C      ADDED RESISTANCE DUE TO INTERSECTION WITH HULL                   FOILSC     17
      C                                                                       FOILSC     18
          RINTS=(XS/SS)*TS2*((.75*TOC)-(.0003/TOC2))                          FOILSC     19
      C                                                                       FOILSC     20
      C      BASE DRAG DUE TO BLUNTNESS OF TRAILING EDGE                      FOILSC     21
      C                                                                       FOILSC     22
          IF(SBS.LE.0.) GO TO 9000                                            FOILSC     23
          RBS=(XS/SS)*SBS*(.135/((CLOTB*CFS)**(1./3.)))                       FOILSC     24
      9000 CONTINUE                                                           FOILSC     25
          RBS=0.                                                              FOILSC     26
          RETURN                                                              FOILSC     27
          END                                                                 FOILSC     28
```

```
      SUBROUTINE FRICT(CF,VF,XLEN,XNU)                                FRICT     2
C                                                                     FRICT     3
C        FLAT PLATE FRICTIONAL COEFFICIENT BASED ON LOCAL REYNOLDS NO.   FRICT     4
C                                                                     FRICT     5
      RE=VF*XLEN/XNU                                                  FRICT     6
      IF(RE.LT.50000.) GO TO 30                                       FRICT     7
      CFTURB=.075/ALOG10(RE/100.)**2                                  FRICT     8
      Y=CFTURB                                                        FRICT     9
   10 CONTINUE                                                        FRICT    10
      CFTURB=.058564/ALOG10(RE*Y)**2                                  FRICT    11
      IF(ABS(CFTURB-Y).LE.1.0E-07) GO TO 20                           FRICT    12
      Y=(CFTURB+Y)/2.                                                 FRICT    13
      GO TO 10                                                        FRICT    14
   20 CONTINUE                                                        FRICT    15
      CF=CFTURB-(143.18/RE)                                           FRICT    16
      RETURN                                                          FRICT    17
   30 CONTINUE                                                        FRICT    18
      CF=1.328/SQRT(RE)                                               FRICT    19
      RETURN                                                          FRICT    20
      END                                                             FRICT    21
```

```
      SUBROUTINE SHPCMP(VDES,EHPDES,PDIA,BDIA,HB,VOFF,EHP,SHP,NV,TITLE) SHPCMP   2
C                                                                       SHPCMP   3
C     ***************************************************************** SHPCMP   4
C        THESE PROGRAMS WHICH DETERMINE A PROPELLER DESIGN AND OFF      SHPCMP   5
C        DESIGN PERFORMANCE ARE EXCERPTS FROM CODING BY ROBERT F. RODDY SHPCMP   6
C        THIS PACKAGE WAS ASSEMBLED IN JUNE 1975 BY WILLIAM G. DAY      SHPCMP   7
C        NSRDC CODE 1524 THE WORK WAS FUNDED UNDER 1-1507-100-60        SHPCMP   8
C     ***************************************************************** SHPCMP   9
C                                                                       SHPCMP  10
      COMMON/CDESC/RHOD(3),ANUD(3),IP                                   SHPCMP  11
C                                                                       SHPCMP  12
      COMMON/CPROP/IPTYP(3),DIAM(3),VADES(3),RPMDES(3),TDES(3),QDES(3), SHPCMP  13
     1        EFFDES(3),OMWTD(3),CAV(3),Z(3),H(3),BAR(3),PD(3),J(3),    SHPCMP  14
     2        IERROR(3),KT(3),KQ(3)                                     SHPCMP  15
C                                                                       SHPCMP  16
      DIMENSION TITLE(8)                                                SHPCMP  17
C                                                                       SHPCMP  18
C        IPTYP  --INDICATE PROPULSOR TYPE                               SHPCMP  19
C        DIAM   --DIAMETER OF PROPULSOR                                 SHPCMP  20
C        VADES  --SPEED OF ADVANCE OF PROPULSOR AT DESIGN VA IN KNOTS   SHPCMP  21
C        RPMDES --RPM AT DESIGN POINT                                   SHPCMP  22
C        TDES   --THRUST OF PROPULSOR SYSTEM IN LBS                     SHPCMP  23
C        QDES   --TORQUE OF PROPULSOR SYSTEM IN FT-LBS                  SHPCMP  24
C        EFFDES --EFFICIENCY OF PROPULSOR                               SHPCMP  25
C        OMWTD  --THRUST WAKE (1-WT)                                    SHPCMP  26
C        CAV    --PERCENT BACK CAVITATION ALLOWED ON WATER SCREWS       SHPCMP  27
C        Z      --NUMBER OF BLADES OF PROPULSOR                         SHPCMP  28
C        H      --DEPTH OF WATER TO CENTER LINE OF PROPULSOR SHAFT      SHPCMP  29
C        BAR    --BLADE AREA RATIO OF PROPULSOR                         SHPCMP  30
C        PD     --P/D OF PROPULSOR                                      SHPCMP  31
C        J      --J OF PROPULSOR                                        SHPCMP  32
C        IERROR --INDICATOR OF ERROR MESSAGE FOR PROPULSOR              SHPCMP  33
C        KT     --KT OF PROPULSOR                                       SHPCMP  34
C        KQ     --KQ OF PROPULSOR                                       SHPCMP  35
C                                                                       SHPCMP  36
      INTEGER Z                                                         SHPCMP  37
C                                                                       SHPCMP  38
      REAL J,KT,KQ                                                      SHPCMP  39
C                                                                       SHPCMP  40
      DIMENSION VOFF(NV),EHP(NV),SHP(NV)                                SHPCMP  41
C                                                                       SHPCMP  42
      IP = 1                                                            SHPCMP  43
      IS=1                                                              SHPCMP  44
      RHOD(IS) = 1.9905                                                 SHPCMP  45
      ANUD(IS) = 0.000012817                                            SHPCMP  46
      CAV(IS) = 2.5                                                     SHPCMP  47
      Z(IS) = 4.                                                        SHPCMP  48
      DIAM(IS) = PDIA                                                   SHPCMP  49
      IPTYP(IS) = 1                                                     SHPCMP  50
      VOF = 0.0                                                         SHPCMP  51
      TOF = 0.0                                                         SHPCMP  52
C                                                                       SHPCMP  53
C        THRUST DEDUCTION (1-T) IS                                      SHPCMP  54
C                                                                       SHPCMP  55
      OMT = 1. - (0.05/(DIAM(IS)/BDIA)**1.26)                           SHPCMP  56
C                                                                       SHPCMP  57
C        THRUST WAKE (1-WT) IS                                          SHPCMP  58
```

```
C                                                                             SHPCMP  59
      OMWT = 1. - (0.71/EXP(1.58*(DIAM(IS) SDIA))                             SHPCMP  60
      OMWTD(IS)=OMWT                                                          SHPCMP  61
      H(IS) = HB                                                              SHPCMP  62
      TDES(IS) = EHPDES*550.0/(VDES*OMT)                                      SHPCMP  63
      VADES(IS) = VDES*OMWT/1.6878                                            SHPCMP  64
C                                                                             SHPCMP  65
C        DETERMINE THE CHARACTERISTICS AND PERFORMANCE OF PROPULSOR           SHPCMP  66
C                                                                             SHPCMP  67
      CALL PRODES(IS)                                                         SHPCMP  68
                                                                              SHPCMP  69
C        PRINT PERFORMANCE DATA OF PROPULSOR                                  SHPCMP  70
C                                                                             SHPCMP  71
      CALL PROPP(IS,TITLE)                                                    SHPCMP  72
C                                                                             SHPCMP  73
      SHPDES = 6.2831853*QDES(IS)*RPMDES(IS)/33000.                           SHPCMP  74
      PC = TDES(IS)*VDES*OMT/(SHPDES*550.0)                                   SHPCMP  75
      SHP(NV) = SHPDES                                                        SHPCMP  76
      WRITE(6,703) SHPDES,PC                                                  SHPCMP  77
      CT=8.0*TDES(IS)/(3.14159*RHOD(IP)*(VADES(IS)*DIAM(IS)*1.6878)**2)       SHPCMP  78
      WRITE(6,704) CT,OMT,OMWT                                                SHPCMP  79
C                                                                             SHPCMP  80
C        CALCULATE OPEN WATER CURVE OF PROPULSOR                              SHPCMP  81
C                                                                             SHPCMP  82
      CALL PROPER(IS,VOF,TOF,DUMMY,TITLE)                                     SHPCMP  83
C                                                                             SHPCMP  84
      DO 22 I=1,NV                                                            SHPCMP  85
         VKOFF = VOFF(I)/1.6878                                               SHPCMP  86
         TOFF = EHP(I)*550.0/(VOFF(I)*OMT)                                    SHPCMP  87
C                                                                             SHPCMP  88
C            DETERMINE OFF DESIGN PERFORMANCE                                 SHPCMP  89
C                                                                             SHPCMP  90
         CALL PROPERO(IS,VKOFF,TOFF,SHP(I),TITLE)                             SHPCMP  91
   22 CONTINUE                                                                SHPCMP  92
      RETURN                                                                  SHPCMP  93
  703 FORMAT(//,1X,*SHP= *,F12.3,5X,*ETA =*,F6.3////)                         SHPCMP  94
C                                                                             SHPCMP  95
  704 FORMAT(20X,*THRUST COEFFICIENT =*,F6.3,/20X,*THRUST DEDUCTION(1-T)      SHPCMP  96
     1 =*,F6.3/20X,*THRUST WAKE  (1-WT) =*,F6.3)                              SHPCMP  97
      END                                                                     SHPCMP  98
```

```
      SUBROUTINE PRODES(IS)                                             PRODES     2
C                                                                       PRODES     3
C  PRODES IS USED IN DESIGN OF PROPULSION SYSTEM TO DETERMINE THE       PRODES     4
C  CHARACTERISTICS AND PERFORMANCE AT DESIGN CONDITION                  PRODES     5
C                                                                       PRODES     6
C  INPUT-OUTPUT PARAMETERS ARE INTRODUCED THROUGH COMMON BLOCK /CPROP/. PRODES     7
C  ARGUMENT -IS- INDICATES PROPULSION SYSTEM BEING CONSIDERED.          PRODES     8
C                                                                       PRODES     9
      COMMON/BARCK/IE                                                   PRODES    10
C                                                                       PRODES    11
      COMMON/CDESC/RHOD(3),ANUD(3),IP                                   PRODES    12
C                                                                       PRODES    13
      COMMON/CPROP/IPTYP(3),DIAM(3),VADES(3),RPMDES(3),TDES(3),QDES(3), PRODES    14
     1       EFFDES(3),OMWTD(3),CAV(3),Z(3),H(3),BAR(3),PD(3),J(3),     PRODES    15
     2       IERROR(3),KT(3),KQ(3)                                      PRODES    16
C                                                                       PRODES    17
      REAL J,KT,KQ                                                      PRODES    18
C                                                                       PRODES    19
      INTEGER Z,TYPIP                                                   PRODES    20
C                                                                       PRODES    21
      IERROR(IS)=0                                                      PRODES    22
      IE=0                                                              PRODES    23
      ITQ=1                                                             PRODES    24
      IPJ=2                                                             PRODES    25
      TYPIP=IPTYP(IS)                                                   PRODES    26
      BAR(IS) = 0.70                                                    PRODES    27
      WRITE(6,600)                                                      PRODES    28
  511 CONTINUE                                                          PRODES    29
C                                                                       PRODES    30
C        OPTIMIZE RPM OR DIAMETER OF PROPULSOR                          PRODES    31
C                                                                       PRODES    32
      CALL TROOST(VADES(IS),QDES(IS),TDES(IS),RPMDES(IS),DIAM(IS),Z(IS),PRODES    33
     1PD(IS),BAR(IS),KT(IS),KQ(IS),EFFDES(IS),J(IS),ITQ,IPJ,RHOD(IP),   PRODES    34
     2IE)                                                               PRODES    35
      IF(IERROR(IS).EQ.3.OR.IERROR(IS).EQ.4) GO TO 515                  PRODES    36
C                                                                       PRODES    37
C            EVALUATE BLADE AREA RATIO                                  PRODES    38
C                                                                       PRODES    39
         CALL BURIL( TDES(IS), DIAM(IS), H(IS), VADES(IS), RPMDES(IS),  PRODES    40
     1      CAV(IS), BAR(IS), IERROR(IS), PD(IS), NRET )                PRODES    41
         IF ( NRET .NE. 0 ) GO TO 511                                   PRODES    42
  515 CONTINUE                                                          PRODES    43
      IERROR(IS)=100*IE+IERROR(IS)                                      PRODES    44
      RETURN                                                            PRODES    45
  600 FORMAT(1H1)                                                       PRODES    46
      END                                                               PRODES    47
```

```
      SUBROUTINE PROPER(IS,VOF,TOF,PWR,TITLE)
C
C  SUBROUTINE "PROPER" CALCULATES THE OPEN WATER CURVES AND OFF-DESIGN
C  PERFORMANCE FOR PROPULSOR SYSTEM "IS"
C
      COMMON/CDESC/RHOD(3),ANUD(3),IP
C
      COMMON/CPROP/IPTYP(3),DIAM(3),VADES(3),RPMDES(3),TDES(3),QDES(3),
     1       EFFDES(3),OMWTD(3),CAV(3),Z(3),H(3),BAR(3),PD(3),J(3),
     2       IERROR(3),KT(3),KQ(3)
C
      COMMON/PCOF/CT(4),CQ(4)
C
      COMMON/ROOTC/A(4)
C
      DIMENSION TITLE(8)
      REAL J,KT,KQ,JK,JOF,KTT,KQQ
C
      INTEGER Z
C
        IJK=0
        PDC=PD(IS)
C
C          EVALUATE THE COEFFICIENTS OF THE KT AND KQ POLYNOMIAL
C
        CALL POLCOF(PDC    ,BAR(IS),Z(IS))
        JK=0.0
C
C          PRINT OPEN WATER CURVE TABLE
C
      WRITE(6,905) TITLE
        WRITE (6,906)
  210   CONTINUE
        KTT    =CT(1)+(CT(2)+(CT(3)+CT(4)*JK    )*JK    )*JK
        KQQ    =CQ(1)+(CQ(2)+(CQ(3)+CQ(4)*JK    )*JK    )*JK
        EEE=KTT*JK/(KQQ*6.2831853)
        WRITE (6,902) JK,KTT,KQQ,EEE
        IF(KTT.LT.0.03) GO TO 970
          JK=JK+0.05
          IF ( JK .LE. 2.0 ) GO TO 210
      GO TO 970
      ENTRY PROPERO
C
C           CALCULATE OFF DESIGN PERFORMANCE
C
        VAFF=VOF*1.6879*OMWTD(IS)
        COEFF=TOF/(RHOD(IP)*DIAM(IS)*DIAM(IS)*VAFF*VAFF)
          DO 950 I=1,4
            A(I)=CT(I)
  950     CONTINUE
        CALL ROOTP(PD(IS),Z(IS),BAR(IS),COEFF,2,1,XKT,XKQ,EEE,JOF)
        XKQ    =CQ(1)+(CQ(2)+(CQ(3)+CQ(4)*JOF   )*JOF   )*JOF
        XKT    =CT(1)+(CT(2)+(CT(3)+CT(4)*JOF   )*JOF   )*JOF
        EEE=XKT*JOF/(XKQ*6.2831853)
        PWR=RHOD(IP)*XKQ*DIAM(IS)*DIAM(IS)*VAFF*VAFF*VAFF/(87.5352188*
     1     JOF*JOF*JOF)
  970 CONTINUE
```

```
      RETURN                                                            PROPER     59
  905 FORMAT(1H1,* TABLE OF OPEN WATER CURVE FOR *,8A10)                PROPER     60
  906 FORMAT(1H0,*             J         KT        KQ          EFF*/)   PROPER     61
  902 FORMAT (5X,F10.2,F10.4,F10.5,F10.3)                               PROPER     62
      END                                                               PROPER     63
```

```
      SUBROUTINE PROPP(IS,TITLE)                                        PROPP      2
C                                                                       PROPP      3
C  THIS ENTRY PRINTS ALL DATA IN COMMON BLOCK "CPROP"                   PROPP      4
C                                                                       PROPP      5
      COMMON/CPROP/IPTYP(3),DIAM(3),VADES(3),RPMDES(3),TDES(3),QDES(3), PROPP      6
     1       EFFDES(3),OMWTD(3),CAV(3),Z(3),H(3),BAR(3),PD(3),J(3),     PROPP      7
     2       IERROR(3),KT(3),KQ(3)                                      PROPP      8
C                                                                       PROPP      9
      DIMENSION TITLE(8)                                                PROPP     10
C                                                                       PROPP     11
      INTEGER Z                                                         PROPP     12
C                                                                       PROPP     13
      REAL J,KT,KQ                                                      PROPP     14
C                                                                       PROPP     15
      WRITE(6,700) TITLE                                                PROPP     16
      WRITE(6,701) VADES(IS),TDES(IS),DIAM(IS),QDES(IS),RPMDES(IS)      PROPP     17
     1,EFFDES(IS),BAR(IS),KT(IS),CAV(IS,,PD(IS),KQ(IS),Z(IS),J(IS),     PROPP     18
     2    IPTYP(IS;,IERROR(IS)                                          PROPP     19
C                                                                       PROPP     20
C         PRINT ERROR MESSAGE                                           PROPP     21
C                                                                       PROPP     22
      IR=IERROR(IS)                                                     PROPP     23
      CALL ERROR(IR)                                                    PROPP     24
      RETURN                                                            PROPP     25
  700 FORMAT(1H1,8A10)                                                  PROPP     26
701   FORMAT(12X,31H   PROPULSOR CHARACTERISTICS    ,                   PROPP     27
     1//1X,18HVELOCITY-(KNOTS) -,F9.2,7X,16HTHRUST-(LBS)    -,F10.0/1X, PROPP     28
     218HDIAMETER-(FEET)  -,F9.2,7X,16HTORQUE-(FT-LBS)-,F10.0,/1X,      PROPP     29
     318HRPM        -        -    ,F8.1,/1X,                            PROPP     30
     511HEFFICIENCY-,F7.3,7H   BAR-,F6.3,7H    KT-,F7.4/1X,             PROPP     31
     611HCAV OR TIP-,F6.1,8H    P/D-,F6.3,7H    KQ-,F8.5/1X,            PROPP     32
     711HNO. BLADES-,I3,7X,4HJ  -,F6.3,//                               PROPP     33
     1X,17HPROPULSOR TYPE  -,I4                                         PROPP     34
     2/  1X,17HERROR TYPE       -,I4)                                   PROPP     35
      END                                                               PROPP     36
```

```
      SUBROUTINE ERROR(IE)                                              ERROR      2
C                                                                       ERROR      3
C        PRINT ERROR MESSAGE OF PROPULSOR                               ERROR      4
C                                                                       ERROR      5
      IF(IE.EQ.0) WRITE(6,100)                                          ERROR      6
      IF(IE.EQ.100) WRITE(6,200)                                        ERROR      7
      IF(IE.EQ.200) WRITE(6,300)                                        ERROR      8
      IF(IE.EQ.3) WRITE(6,400)                                          ERROR      9
      IF(IE.EQ.4) WRITE(6,500)                                          ERROR     10
      IF(IE.EQ.5) WRITE(6,600)                                          ERROR     11
      IF(IE.EQ.6) WRITE(6,700)                                          ERROR     12
      RETURN                                                            ERROR     13
  100 FORMAT(1H+,25X,*NO ERROR*)                                        ERROR     14
  200 FORMAT(1H+,25X,*LIMIT VALUE FOR P/D HAS BEEN USED*)               ERROR     15
  300 FORMAT(1H+,25X,*ITTERATION MALFUNCTION--PD(2) USED FOR P/D*)      ERROR     16
  400 FORMAT(1H+,25X,*MAX BAR=.95 USED--MARINE SCREW*)                  ERROR     17
  500 FORMAT(1H+,25X,*MIN BAR=.4 USED--MARINE SCREW*)                   ERROR     18
  600 FORMAT(1H+,25X,*MAX TIP SPEED USED--AIR SCREW*)                   ERROR     19
  700 FORMAT(1H+,25X,*BLADE NUMBER OUTSIDE PROGRAM LIMIT--CLOSEST NUMBER ERROR     20
     1 USED*)                                                           ERROR     21
      END                                                               ERROR     22
```

```
      SUBROUTINE TROOST(VA1,P,T,N1,D,Z,PDNEW,BAR,KT,KQ,EFFNEW,J,ITQ,IPJ, TROOST   2
     1RHO,IERROR)                                                        TROOST   3
C                                                                        TROOST   4
C  THIS SUBROUTINE IS TO BE USED WITH SUBROUTINE ROOT AND FUNCTION CALC. TROOST   5
C  EITHER RPM OR DIAMETER CAN BE OPTIMIZED GIVEN THE OTHER(SEE VARIABLE  TROOST   6
C  "IPJ" BELOW)                                                          TROOST   7
C                                                                        TROOST   8
C  IF J,PD,ETC. ARE KNOWN AND ONLY KT,KQ AND EFF ARE NEEDED THE SECOND   TROOST   9
C  ENTRY TO "TROOST" MAY BE USED.  THIS IS "TROST2"                      TROOST  10
C  IF OPEN WATER CURVES ARE DESIRED THIS ENTRY POINT LENDS ITSELF EASILY TROOST  11
C  TO GENERATING THESE CURVES.                                           TROOST  12
C  EXPLAINATION OF VARIABLES IN THE SUBROUTINE STATEMENT                 TROOST  13
C                                                                        TROOST  14
C       VA1    -WAKED SHIP VELOCITY (KNOTS)                              TROOST  15
C       P      -TORQUE PER SHAFT(FT-LBS)                                 TROOST  16
C       T      -PROPELLER THRUST (POUNDS)  PER SHAFT                     TROOST  17
C       N1     -PROPELLER RPM                                            TROOST  18
C       D      -PROPELLER DIAMETER (FEET)                                TROOST  19
C       Z      -NUMBER OF BLADES PER PROPELLER                           TROOST  20
C       PD1    -PITCH (AT 0.7 R) IN FEET DIVIDED BY DIAMETER (IE. P/D)   TROOST  21
C       BAR    -BLADE AREA RATIO (EXPANDED AREA/DISC AREA)               TROOST  22
C       KT     -THRUST COEFFICIENT (KT=T/(RHO*N**2*D**4) )               TROOST  23
C       KQ     -TORQUE COEFFICIENT (KQ=Q/(RHO*N**2*D**5) )               TROOST  24
C       EFF1   -OPEN WATER EFFICIENCY (EFF=J*KT/(2*PI*KQ) )              TROOST  25
C       J      -ADVANCE COEFFICIENT(J=VA/(N*D) )                         TROOST  26
C       ITQ    -INDEXING COEFFICIENT-CAN EQUAL ONLY 1 OR 2               TROOST  27
C              1--THRUST IS AN INPUT                                     TROOST  28
C              2--POWERT IS AN INPUT                                     TROOST  29
C       IPJ    -INDEXING COEFFICIENT-CAN EQUAL 1 THROUGH 5               TROOST  30
C              1--NO OPTIMIZATION,J KNOWN,CALCULATES KT,KQ,EFF1 ONLY     TROOST  31
C              2--OPTIMIZES N WITH KT/J**2                               TROOST  32
C              3--OPTIMIZES N WITH KQ/J**3                               TROOST  33
C              4--OPTIMIZES D WITH KT/J**4                               TROOST  34
C              5--OPTIMIZES D WITH KQ/J**5                               TROOST  35
C       RHO    -WATER DENSITY FOR PROPELLER OPERATION(LB-SEC**2/FT**4)   TROOST  36
C                                                                        TROOST  37
      REAL N,J,KT,KQ,N1                                                  TROOST  38
C                                                                        TROOST  39
      INTEGER Z                                                          TROOST  40
C                                                                        TROOST  41
      DIMENSION EFF(5),PD(5),S(2),COEFF(5)                               TROOST  42
C                                                                        TROOST  43
      IF(Z.GT.7.OR.Z.LT.2) IERROR=6                                      TROOST  44
      IF(Z.GT.7) Z=7                                                     TROOST  45
      IF(Z.LT.2) Z=2                                                     TROOST  46
      N=N1/60.                                                           TROOST  47
      VA=VA1*1.6878                                                      TROOST  48
      IF(IPJ.EQ.1) GO TO 21                                              TROOST  49
         S(1)=T                                                          TROOST  50
         S(2)=P*87.5352188                                               TROOST  51
         IF(IPJ.GE.4) COEFF(IPJ)=S(ITQ)*N*N/(RHO*VA**IPJ)                TROOST  52
         IF(IPJ.LE.3) COEFF(IPJ)=S(ITQ)/(RHO*D*D*VA**IPJ)                TROOST  53
         M=0                                                             TROOST  54
         DEL PD=.05                                                      TROOST  55
         PD(3)=0.40                                                      TROOST  56
C                                                                        TROOST  57
C           FIRST FIND 3 POINTS WHICH CONTAIN THE MAXIMUM EFFICIENCY     TROOST  58
```

```
      C                                                                 TROOST     59
      40     CONTINUE                                                   TROOST     60
             DO 50 I=1,2                                                TROOST     61
                PD(I)=PD(I+1)                                           TROOST     62
                EFF(I)=EFF(I+1)                                         TROOST     63
      50     CONTINUE                                                   TROOST     64
             PD(3)=PD(2)+DEL PD                                         TROOST     65
             IF(PD(3).GT.1.4) GO TO 97                                  TROOST     66
                M=M+1                                                   TROOST     67
                CALL ROOT(PD(3),Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFFIN,J) TROOST     68
                EFF(3)=EFFIN                                            TROOST     69
                IF(M.LE.2.OR.EFF(3).GE.EFF(2)) GO TO 40                 TROOST     70
      C                                                                 TROOST     71
      C            ITERATE TO MAXIMUM EFFICIENCY                        TROOST     72
      C            THIS METHOD IS REFERENCED IN--                       TROOST     73
      C            "INTRODUCTION TO NUMERICAL ANALYSIS" BY F.B. HILDEBRAND  TROOST 74
      C            MC GRAW-HILL 1956, PAGES 50 AND 56.                  TROOST     75
      C                                                                 TROOST     76
                PD(4)=PD(2)-DELPD/2.0                                   TROOST     77
                PD(5)=PD(2)+DELPD/2.0                                   TROOST     78
                CALL ROOT(PD(4),Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFF(4),J)  TROOST   79
                CALL ROOT(PD(5),Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFF(5),J)  TROOST   80
                PDNEW=PD(2)-0.0175*(-EFF(1)-0.5*EFF(4)+0.5*EFF(5)+EFF(3))/  TROOST 81
           1       ( EFF(1)-0.5*EFF(4)-EFF(2)-0.5*EFF(5)+EFF(3))        TROOST     82
                IF(PDNEW.LT.PD(1).OR.PDNEW.GT.PD(3)) GO TO 96           TROOST     83
      C                                                                 TROOST     84
      C               FINAL COMPUTED VALUES OF KT AND KQ ARE COMMON WITH   TROOST  85
      C                  SUBROUTINE ROOT AND HAVE ALREADY BEEN CALCULATED  TROOST  86
      C                                                                 TROOST     87
                   CALL ROOT(PD NEW,Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFFNEW,J) TROOST 88
                GO TO 960                                               TROOST     89
      96        CONTINUE                                                TROOST     90
      C                                                                 TROOST     91
      C               ITTERATION MALFUNCTION WITH PDNEW OUTSIDE THE LIMITS  TROOST 92
      C                  OF PD(1) OR PD(3).  THE VALUE AT PD(2) HAS BEEN    TROOST 93
      C                  USED.                                          TROOST     94
      C                                                                 TROOST     95
                   IERROR = 2                                           TROOST     96
                   PDNEW=PD(2)                                          TROOST     97
                   CALL ROOT(PDNEW,Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFFNEW,J) TROOST 98
      960       CONTINUE                                                TROOST     99
             GO TO 100                                                  TROOST    100
      97     CONTINUE                                                   TROOST    101
      C                                                                 TROOST    102
      C            AT THIS POINT P/D=1.4 HAS BEEN USED AND THE SEARCH FOR   TROOST 103
      C               THE MAXIMUM HAS BEEN TERMINATED                   TROOST    104
      C                                                                 TROOST    105
                IERROR = 1                                              TROOST    106
                EFFNEW=EFF(2)                                           TROOST    107
                PDNEW=PD(2)                                             TROOST    108
      100    CONTINUE                                                   TROOST    109
             IF(IPJ.LE.3) N1=VA1*101.268/(D*J)                          TROOST    110
             IF(IPJ.GE.4) D=VA1*1.6878/((N1/60.0)*J)                    TROOST    111
             P=KQ*RHO*N1*N1*D**5/3600.                                  TROOST    112
          WRITE (6,988) VA1,J,PDNEW                                     TROOST    113
          GO TO 30                                                      TROOST    114
      21  CONTINUE                                                      TROOST    115
```

```
              PDX=0.0
              DPD=0.2
              PDY=1.72
              J=VA/(N*D)
              KT=T/(RHO*N*N*D*D*D*D)
              XY=CALC(J,BAR,PDY,Z,1)
           WRITE(6,989)
           WRITE(6,990) VA1,J,KT,PDY
              IF(KT.GT.XY) GO TO 28
                 PDX=PDY
      22         CONTINUE
                 PDX=PDX-DPD
                 XX=CALC(J,BAR,PDX,Z,1)
           WRITE(6,990) VA1,J,KT,PDY,PDX,XX
                 IF(KT.LT.XX) GO TO 22
      23         CONTINUE
                 DPD=(PDY-PDX)*(KT-XY)/(XY-XX)
                 XX=XY
                 PDX=PDY
                 PDY=PDY+DPD
                 XY=CALC(J,BAR,PDY,Z,1)
           WRITE(6,990) VA1,J,KT,PDY,PDX,XX,XY
                 IF(ABS(KT-XY).GT.0.0001) GO TO 23
                 PDNEW=PDY
                 CALL TROST2(KT,KQ,J,PDNEW,BAR,Z,EFFNEW)
                 IERROR=0
              GO TO 29
      28      CONTINUE
      C
      C             AT THIS POINT P/D=1.4 HAS BEEN USED
      C
                 PDNEW = 1.4
                 IERROR = 1
                 CALL TROST2(KT,KQ,J,PDNEW,BAR,Z,EFFNEW)
      29      CONTINUE
              P=KQ*RHO*N*N*D*D*D*D*D
      30   CONTINUE
           RETURN
  988      FORMAT(1H ,*  FIRST LOOP - ROOT CALL*,8F10.3)
  989      FORMAT(//1H ,* SECOND LOOP - CALC CALL*)
      990  FORMAT(20X,6(F10.3,2X))
           END
```

```
      SUBROUTINE TROST2(KT,KQ,J,PDTR2,BAR,Z,EFFTR2)                     TROST2     2
C                                                                       TROST2     3
C         THIS SUBROUTINE IS USED TO CALCULATE A SINGLE KT, KQ AND      TROST2     4
C           EFFICIENCY VALUE FOR A GIVEN PROPELLER                      TROST2     5
C                                                                       TROST2     6
      REAL  KT,KQ,J                                                     TROST2     7
C                                                                       TROST2     8
      INTEGER Z                                                         TROST2     9
C                                                                       TROST2    10
      KT=CALC(J,BAR,PDTR2,Z,1)                                          TROST2    11
      KQ=CALC(J,BAR,PDTR2,Z,2)                                          TROST2    12
      EFFTR2=KT*J/(KQ*6.2831853)                                        TROST2    13
      RETURN                                                            TROST2    14
      END                                                               TROST2    15
```

```
      REAL FUNCTION CALC(J,BAR,PD,Z,ITQ)                               CALC    2
C THIS FUNCTION CALCULATES"KT"OR"KQ" FOR 2 -- 7 BLADED PROPELLERS GIVEN CALC   3
C J,BAR,P/D,NO. OF BLADES,AND AN INDICATOR FOR "KT" OR "KQ"             CALC    4
C ---REFERENCE---                                                       CALC    5
C   "THE WAGENINGEN B-SCREW SERIES" BY VAN LAMMEREN,VAN MANEN, AND      CALC    6
C    OOSTERVELD    SNAME TRANSACTIONS/VOL. 77/ 1969                     CALC    7
C                        AND                                            CALC    8
C    "RECENT DEVELOPMENTS IN MARINE PROPELLER HYDRODYNAMICS"  BY        CALC    9
C    OOSTERVELD AND OOSSANEN/ PROCEEDINGS OF THE NSMB 40TH ANNIVERSARY/ CALC   10
C    1972                                                               CALC   11
C                                                                       CALC   12
      COMMON/OFFDES/PJAY(4)                                             CALC   13
C                                                                       CALC   14
      REAL J                                                            CALC   15
C                                                                       CALC   16
      INTEGER Z                                                         CALC   17
C                                                                       CALC   18
      DIMENSION C(47,2),IS(47,2),IT(47,2),IU(47,2),IV(47,2),COFJAY(4)   CALC   19
C                                                                       CALC   20
      DATA C/      .880496E-2,-.204554   , .166351   , .158114   ,      CALC   21
     1-.147581   ,-.481497   , .415437   , .144043E-1,-.530054E-1,      CALC   22
     1 .143481E-1, .606826E-1,-.125894E-1, .109689E-1,-.133698   ,      CALC   23
     1 .638407E-2,-.132718E-2, .168496   ,-.507214E-1, .854559E-1,      CALC   24
     1-.504475E-1, .10465 E-1,-.648272E-2,-.841728E-2, .168424E-1,      CALC   25
     1-.102296E-2,-.317791E-1, .18604 E-1,-.410798E-2,-.606848E-3,      CALC   26
     1-.49819 E-2, .25983 E-2,-.560528E-3,-.163652E-2,-.328787E-3,      CALC   27
     1 .116502E-3, .690904E-3, .421749E-2, .565229E-4,-.146564E-2,      CALC   28
     10.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,                              CALC   29
     2 .379368E-2, .886523E-2,-.32241 E-1, .344778E-2,-.408811E-1,      CALC   30
     2-.108009   ,-.885381E-1, .188561   ,-.370871E-2, .513696E-2,      CALC   31
     2 .209449E-1, .474319E-2,-.723408E-2, .438388E-2,-.269403E-1,      CALC   32
     2 .558082E-1, .161886E-1, .318086E-2, .15896 E-1, .471729E-1,      CALC   33
     2 .196283E-1,-.502782E-1,-.30055 E-1, .417122E-1,-.397722E-1,      CALC   34
     2-.350024E-2,-.106854E-1, .110903E-2,-.313912E-3, .35895 E-2,      CALC   35
     2-.142121E-2,-.383637E-2, .126803E-1,-.318278E-2, .334268E-2,      CALC   36
     2-.183491E-2, .112451E-3,-.297228E-4, .269551E-3, .83265 E-3,      CALC   37
     2 .155334E-2, .302683E-3,-.1843  E-3,-.425399E-3, .869243E-4,      CALC   38
     2-.4659  E-3, .554194E-4/                                          CALC   39
C                                                                       CALC   40
      DATA IS/0,1,0,0,2,1,0,0,2,0,1,0,1,0,0,2,3,0,2,3,1,2,0,1,3,0,1,0,0, CALC   41
     11,2,3,1,1,2,0,0,3,0,0,0,0,0,0,0,0,0,                              CALC   42
     20,2,1,0,0,1,2,0,1,0,1,2,2,1,0,3,0,1,0,1,3,0,3,2,0,0,3,3,0,3,0,1,0, CALC   43
     22,0,1,3,3,1,2,0,0,0,0,0,3,0,1/                                    CALC   44
C                                                                       CALC   45
      DATA IT/0,0,1,2,0,1,2,0,0,1,1,0,0,3,6,6,0,0,0,0,6,6,3,3,3,3,0,2,0, CALC   46
     10,0,0,2,6,6,0,3,6,3,0,0,0,0,0,0,0,0,                              CALC   47
     20,0,1,2,1,1,1,2,0,1,1,1,0,1,2,0,3,3,0,0,0,1,1,2,3,6,0,3,6,0,6,0,2, CALC   48
     23,6,1,2,6,0,0,2,6,0,3,3,6,6/                                      CALC   49
C                                                                       CALC   50
      DATA IU/0,0,0,0,1,1,1,0,0,0,0,1,1,0,0,0,1,2,2,2,2,2,0,0,0,1,2,2,0, CALC   51
     10,0,0,0,0,0,1,1,1,2,0,0,0,0,0,0,0,0,                              CALC   52
     20,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2,2,2,0,0,0,1,1,2,2, CALC   53
     22,2,0,0,0,1,1,1,1,2,2,2,2,2/                                      CALC   54
C                                                                       CALC   55
      DATA IV/0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2, CALC   56
     12,2,2,2,2,2,2,2,2,2,0,0,0,0,0,0,0,0,                              CALC   57
     20,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1, CALC   58
```

```
     21,1.2,2,2,2,2,2,2,2,2,2,2,2/                                     CALC     59
C                                                                      CALC     60
         IJK=1                                                         CALC     61
      GO TO 10                                                         CALC     62
      ENTRY CALC1                                                      CALC     63
         IJK=2                                                         CALC     64
10    CONTINUE                                                         CALC     65
      DO 25 I=1,4                                                      CALC     66
         COFJAY(I)=0.0                                                 CALC     67
25    CONTINUE                                                         CALC     68
      X=FLOAT(Z)                                                       CALC     69
      INDEX=39                                                         CALC     70
      IF(ITQ.EQ.2) INDEX=47                                            CALC     71
      DO 100 N = 1, INDEX                                              CALC     72
         DO 50 I = 1, 4                                                CALC     73
            IF(IS(N,ITQ).EQ.I-1) COFJAY(I)= COFJAY(I)+                 CALC     74
     1         C(N,ITQ)*BAR**IU(N,ITQ)*PD**IT(N,ITQ)*X**IV(N,ITQ)      CALC     75
50       CONTINUE                                                      CALC     76
100   CONTINUE                                                         CALC     77
      IF(J.EQ.0.0) GO TO 150                                           CALC     78
         CALC=COFJAY(1)+(COFJAY(2)+(COFJAY(3)+COFJAY(4)*J)*J)*J        CALC     79
      GO TO 200                                                        CALC     80
150   CONTINUE                                                         CALC     81
         CALC=COFJAY(1)                                                CALC     82
200   CONTINUE                                                         CALC     83
      IF(IJK.NE.2) GO TO 250                                           CALC     84
         DO 225 I=1,4                                                  CALC     85
            PJAY(I)=COFJAY(I)                                          CALC     86
225      CONTINUE                                                      CALC     87
250   CONTINUE                                                         CALC     88
      RETURN                                                           CALC     89
      END                                                              CALC     90
```

```
      SUBROUTINE ROOT(PD,Z,BAR,COEFF,IPJ,ITQ,KT,KQ,EFFIN,J2)
C
C        THIS SUBROUTINE USES THE REGULA FALSI METHOD TO FIND THE
C           INTERSECTION OF THE "K" CURVE AND THE "K/J**POWER" CURVE.
C        ---REFERENCE---
C           "INTRODUCTION TO NUMERICAL ANALYSIS" BY F.B. HILDEBRAND
C           MCGRAW-HILL  1956  PAGE 446
C
      COMMON/ROOTC/A(4)
C
      REAL KT,KQ,J2
C
      INTEGER Z
C
         IJK=1
      GO TO 5
      ENTRY ROOTP
         IJK=2
    5 CONTINUE
      I = 1
      EPS = 0.00005
      J2=0.0
      IF(ITQ.EQ.2) EPS = 0.000005
      IF(IJK.EQ.1) GO TO 22
         TQ1= A(1)+(A(2)+(A(3)+A(4)*J2)*J2)*J2
      GO TO 23
   22 CONTINUE
         TQ1=CALC(J2,BAR,PD,Z,ITQ)
   23 CONTINUE
      DEL=(TQ1/COEFF)**(1.0/FLOAT(IPJ))
      IF(PD.LT.DEL) DEL =PD
      DELL=DEL/2.0
      IF(IJK.EQ.1) GO TO 24
         TQ3=(A(1)+(A(2)+(A(3)+A(4)*DEL )*DEL )*DEL )-COEFF*DEL **IPJ
         TQ =(A(1)+(A(2)+(A(3)+A(4)*DELL)*DELL)*DELL)-COEFF*DELL**IPJ
      GO TO 25
   24 CONTINUE
         TQ3=CALC(DEL,BAR,PD,Z,ITQ)-COEFF*DEL**IPJ
         TQ=CALC(DELL,BAR,PD,Z,ITQ)-COEFF*DELL**IPJ
   25 CONTINUE
      IF(TQ.LT.0.0) GO TO 30
         TQ2=TQ
         TQ1=TQ3
         J2=DEL
         DEL=DELL
      GO TO 515
   30 CONTINUE
         DEL=DELL
   51    CONTINUE
         TQ2=TQ1
         J2=J2+DEL
         IF(IJK.EQ.1) GO TO 512
            TQ1= A(1)+(A(2)+(A(3)+A(4)*J2)*J2)*J2
         GO TO 513
  512    CONTINUE
            TQ1=CALC(J2,BAR,PD,Z,ITQ)
  513    CONTINUE
```

```
          TQ1=TQ1-COEFF*J2**IPJ
  515  CONTINUE
       IF(ABS(TQ1)-EPS.LE.0.) GO TO 52
          DE=DEL*TQ1/(TQ2-TQ1)
          IF(ABS(DE).LT.ABS(DEL)) GO TO 518
             DEL=SIGN(DEL,DE)
          GO TO 519
  518     CONTINUE
             DEL=DE
  519     CONTINUE
          I=I+1
          GO TO 51
  52   CONTINUE
       IF(IJK.EQ.2) GO TO 53
          KT=CALC(J2,BAR,PD,Z,1)
          KQ=CALC(J2,BAR,PD,Z,2)
          EFFIN=KT*J2/(KQ*6.2831853)
  53   CONTINUE
       RETURN
       END
```

```
      SUBROUTINE BURIL(  T,D,H,VA,N,CAVI,BAR,IERROR,PD,NRET)
C
C        THIS SUBROUTINE USES BURRILL"S CAVITATION CRITERIA TO DETERMINE
C          IF THE BAR USED IS SUFFICIENT AND, IF IT IS NOT, TO ESTIMATE
C                WHAT BAR IS NECESSARY
C
      COMMON/BARCK/IE
C
      REAL N
C
      NRET = 0
      BARC=BARB
      BARB=BAR
      THRUST=T
      AO=0.7853975*D*D
      DELTP=14.45+0.45*H
      QSUBT=(VA/7.12)**2+(N*D/329.0)**2
      SIGM=DELTP/QSUBT
      TAU=0.17096999E-1
     1+.23641524*SIGM
     1-.52794965E-2*CAVI
     1+.78910550E-1*SIGM*CAVI
     1+.42061131E-5*SIGM*CAVI**3
     1-.98059874E-1*SIGM**2*CAVI
     1-.40621894E-2*SIGM**2*CAVI**2
     1-.18207486E-7*SIGM**2*CAVI**5
     1+.75936652E-2*SIGM**3*CAVI
     1+.92023581E-2*SIGM**3*CAVI**2
     1-.17649923E-3*SIGM**5*CAVI**2
     1-.34804349E-4*SIGM**5*CAVI**4
     1+.99981727E-6*SIGM**5*CAVI**5
      AP=THRUST/(TAU*QSUBT*144.0)
      PAR=AP/AO
      EAR=PAR/(1.067-0.229*PD)
      IF(ABS(BAR-EAR).LE.0.0005) GO TO 90
      IF(IE.EQ.1.AND.ABS(EAR-BARC).LE.0.01) GO TO 95
         BAR=EAR
         IF(BAR.GT.0.95) GO TO 70
         IF(BAR.LT.0.4) GO TO 80
         NRET = 1
         GO TO 95
   70    CONTINUE
C
C           EAR IS GREATER THAN 0.95, USE 0.95 FOR EAR
C
         BAR = 0.95
         IERROR = 3
         NRET = 1
         GO TO 95
   80    CONTINUE
C
C           EAR IS LESS THAN 0.4 USE 0.4 FOR BAR
C
         BAR=0.4
         IERROR = 4
         NRET = 1
         GO TO 95
```

```
      90   CONTINUE                                    BURIL     59
           IERROR = 0                                  SURIL     60
      95   CONTINUE                                    BURIL     61
           RETURN                                      BURIL     62
           END                                         BURIL     63
```

```
      SUBROUTINE POLCOF(PD,BAR,Z)
C
C        DETERMINE THE COEFFICIENTS OF THE POLYNOMIALS WHICH APPROXIMATE
C              KT AND KQ CURVE
C
      COMMON/OFFDES/PJAY(4)
C
      COMMON/PCOF/CT(4),CQ(4)
C
      REAL KT,KQ,J
C
      INTEGER Z
C
      J=0.0
      KT=CALC1(J,BAR,PD,Z,1)
      DO 5 I=1,4
         CT(I)=PJAY(I)
    5 CONTINUE
      KQ=CALC1(J,BAR,PD,Z,2)
      DO 7 I=1,4
         CQ(I)=PJAY(I)
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BESSJ(X,N,VJ)                                          BESSJ   2
C                                                                       BESSJ   3
C        EVALUATE BSEESL FUNCTION FROM ORDER 0 TO ORDER N               BESSJ   4
C                                                                       BESSJ   5
      REAL J, JL, JLL, JLLL                                             BESSJ   6
      DIMENSION VJ(N)                                                   BESSJ   7
      DATA PI /3.1415926535898/,                                        BESSJ   8
     1     EPS /1.0E-30/,                                               BESSJ   9
     2     EXPON /2.7182818284590/                                      BESSJ  10
C                                                                       BESSJ  11
C        DETERMINE THE MAXIMUM N = NU                                   BESSJ  12
C                                                                       BESSJ  13
      D = ALOG(EPS)/X                                                   BESSJ  14
      C = ALOG(EXPON/2.0)                                               BESSJ  15
      B = D-2.0-C                                                       BESSJ  16
      ALPHA = (-B+SQRT(B**2-4.0*D*(2.0-C)))/(2.0*(2.0-C))               BESSJ  17
      CLN = ALOG(2.0*PI*X)                                              BESSJ  18
      IF(ALPHA.LT.1.5) ALPHA = 1.5                                      BESSJ  19
      DO 1000 I=1,20                                                    BESSJ  20
         ALN = ALOG(ALPHA)                                              BESSJ  21
         F = -(CLN+ALN)/(2.0*X)+ALPHA*(C-ALN)-D                         BESSJ  22
         FP = -1.0/(2.0*X*ALPHA)+C-1.0-ALN                              BESSJ  23
         DEL = F/FP                                                     BESSJ  24
      ALPHA = ALPHA-DEL                                                 BESSJ  25
         IF(X*ABS(DEL) .LE. 0.01) GO TO 1100                            BESSJ  26
 1000 CONTINUE                                                          BESSJ  27
      GO TO 88888                                                       BESSJ  28
 1100 CONTINUE                                                          BESSJ  29
      GNU = ALPHA*X                                                     BESSJ  30
      NU = IFIX(GNU)+1                                                  BESSJ  31
      IF(NU.LT.N+10) NU = N+10                                          BESSJ  32
      IF((NU/2)*2.NE.NU) NU=NU+1                                        BESSJ  33
C                                                                       BESSJ  34
C         RECURSE IN PLACE FOR M.GT.N                                   BESSJ  35
C                                                                       BESSJ  36
      SUM = 0.0                                                         BESSJ  37
      JLLL = 0.0                                                        BESSJ  38
      JLL = EPS                                                         BESSJ  39
      NUM1 = NU-1                                                       BESSJ  40
      NP = N                                                            BESSJ  41
      IF((NP/2)*2.EQ.NP) NP = NP-1                                      BESSJ  42
      DO 1200 MM = NP,NUM1,2                                            BESSJ  43
      M = NUM1-MM+NP                                                    BESSJ  44
         JL = 2.0*FLOAT(M+2)*JLL/X-JLLL                                 BESSJ  45
         J = 2.0*FLOAT(M+1)*JL/X-JLL                                    BESSJ  46
         SUM = SUM+2.0*JL                                               BESSJ  47
         JLLL = JL                                                      BESSJ  48
         JLL = J                                                        BESSJ  49
 1200 CONTINUE                                                          BESSJ  50
      IF(NP.NE.N) GO TO 1300                                            BESSJ  51
      JL = 2.0*FLOAT(N)*JLL/X-JLLL                                      BESSJ  52
      SUM = SUM+2.0*JL                                                  BESSJ  53
      JLLL = JLL                                                        BESSJ  54
      JLL = JL                                                          BESSJ  55
 1300 CONTINUE                                                          BESSJ  56
      I = N+1                                                           BESSJ  57
      VJ(I) = JLLL                                                      BESSJ  58
```

```
      VJ(N) = JLL
      NP = N-1
      DO 1400 MM = 1,NP
      M = NP-MM
      K = M+1
      VJ(K) = 2.0*FLOAT(K)*VJ(M+2)/X-VJ(M+3)
      IF((M/2)*2.EQ.M) SUM = SUM+2*VJ(M+1)
 1400 CONTINUE
      SUM = SUM-VJ(1)
      K = N+1
      DO 1500 M = 1,K
      VJ(M) = VJ(M)/SUM
 1500 CONTINUE
99999 CONTINUE
      RETURN
88888 CONTINUE
      WRITE(6,100) X, ALPHA, DEL
  100 FORMAT(28H NO CONVERGENCE, X,ALPHA,DEL/3E14.7)
      END
```

```
      SUBROUTINE PCHEB (AS, BS, AB, BB, NN, TITLE)                      PCHEB     2
C                                                                       PCHEB     3
C     PLOT BODY SECTIONAL AREA CURVE AND STRUT WATERPLANE OUTLINE CURVE PCHEB     4
C     INPUTS                                                            PCHEB     5
C         AS AND BS ARE, RESPECTIVELY, THE VECTORS OF SYMMETRIC-        PCHEB     6
C         AND OF ANTISYMMETRIC-CHEBYCHEV COEFFICIENTS FOR THE STRUT.    PCHEB     7
C                                                                       PCHEB     8
C         AB AND BB ARE, RESPECTIVELY, THE VECTORS OF SYMMETRIC-        PCHEB     9
C         AND OF ANTISYMMETRIC-CHEBYCHEV COEFFICIENTS FOR THE BODY.     PCHEB    10
C                                                                       PCHEB    11
C         NN = DIMENSION OF AS, BS, AB, BB                              PCHEB    12
C           MAXIMUM VALUE OF NN IS NMAX                                 PCHEB    13
C           NMAX = 10                                                   PCHEB    14
C                                                                       PCHEB    15
C                                                                       PCHEB    16
C                                                                       PCHEB    17
C                                                                       PCHEB    18
C     INITIALIZE THE GRID GEOMETRY, THE PLOT VARIABLES AND              PCHEB    19
C     THE ABSCISSA VARIABLE.                                            PCHEB    20
C                                                                       PCHEB    21
      COMMON/PLOT/NFIRST,NLAST,NPOINT,XMAX,XMIN,NSCL1,NCHAR,NSCALE(4),  PCHEB    22
     1            PCHAR(2)                                              PCHEB    23
      COMMON/XRPLOTQ/II,JJ,KK,LL,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,         PCHEB    24
     1               ISX,ISY,V,H                                        PCHEB    25
      LOGICAL NSCL1                                                     PCHEB    26
C                                                                       PCHEB    27
      DIMENSION AS(NN), BS(NN), AB(NN), BB(NN), TITLE(8)                PCHEB    28
      DIMENSION STAT(101),YS(101),YB(101),LABLE(3)                      PCHEB    29
C                                                                       PCHEB    30
C                                                                       PCHEB    31
C    GENERATE ORDINATE VALUES FOR STRUT AND BODY POINTS                 PCHEB    32
C                                                                       PCHEB    33
      DO 2000 I=1, 101                                                  PCHEB    34
         STAT(I)=2.0*(FLOAT(I-1)*.01-.5)                                PCHEB    35
         THETA=ASIN(STAT(I))                                            PCHEB    36
         YS(I)=0.0                                                      PCHEB    37
         YB(I)=0.0                                                      PCHEB    38
         DO 1000 NCO=1,NN                                               PCHEB    39
            UM=COS(FLOAT(2*NCO-1)*THETA)                                PCHEB    40
            VM=SIN(FLOAT(2*NCO)*THETA)                                  PCHEB    41
            YS(I)=YS(I)+AS(NCO)*UM+BS(NCO)*VM                           PCHEB    42
            YB(I)=YB(I)+AB(NCO)*UM+BB(NCO)*VM                           PCHEB    43
 1000    CONTINUE                                                       PCHEB    44
 2000 CONTINUE                                                          PCHEB    45
C                                                                       PCHEB    46
C     DETERMINE THE MINIMUM AND MAXIMUM ORDINATE VALUES                 PCHEB    47
C                                                                       PCHEB    48
      YMAX = -999999.0                                                  PCHEB    49
      YMIN = +999999.0                                                  PCHEB    50
      DO 3000 J=1,101                                                   PCHEB    51
         IF(YMAX .LT. YB(J)) YMAX=YB(J)                                 PCHEB    52
         IF(YMAX .LT. YS(J)) YMAX=YS(J)                                 PCHEB    53
         IF(YMIN .GT. YB(J)) YMIN=YB(J)                                 PCHEB    54
         IF(YMIN .GT. YS(J)) YMIN=YS(J)                                 PCHEB    55
 3000 CONTINUE                                                          PCHEB    56
C                                                                       PCHEB    57
C     CALL PLOT1 TO SET UP GRID SPACING AND DETERMINE THE AXIS VALUES   PCHEB    58
```

```
      C                                                                 PCHEB    59
            CALL PLOT1 (NSCALE,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR)           PCHEB    60
      C                                                                 PCHEB    61
      C     CALL PLOT2 TO EXAMINE THE MINIMUM AND MAXIMUM VALUES OF     PCHEB    62
      C     ABSCISSA AND ORDINATE AND TO ESTABLISH AN INTERNAL FORMULA  PCHEB    63
      C     FOR COMPUTING THE LOCATION IN THE IMAGE REGION CORRESPONDING PCHEB   64
      C     TO THE POINT TO BE PLOTTED.                                 PCHEB    65
      C     WHEN NSCL1=.TRUE., THE STANDARD GRID WILL NOT BE USED       PCHEB    66
      C                                                                 PCHEB    67
            CALL PLOT2 (XMAX,XMIN,YMAX,YMIN,NSCL1)                      PCHEB    68
      C                                                                 PCHEB    69
      C     CALL PLOT3 FOR ASSIGNING AN ALPHA-CHARACTER TO EACH POINT   PCHEB    70
      C     WHICH WILL BE PLOTTED.                                      PCHEB    71
      C                                                                 PCHEB    72
            CALL PLOT3 (PCHAR(1),STAT,YS,NFIRST,NLAST,NPOINT)           PCHEB    73
            CALL PLOT3 (PCHAR(2),STAT,YB,NFIRST,NLAST,NPOINT)           PCHEB    74
      C                                                                 PCHEB    75
      C     PRINT THE HEADING OF THE PLOT                               PCHEB    76
      C                                                                 PCHEB    77
            WRITE (6,9000)  (TITLE(I), I=1,8)                           PCHEB    78
      C                                                                 PCHEB    79
      C     CALL PLOT4 TO PRINT THE IMAGE OF THE COMPLETED GRAPH ON THE PCHEB    80
      C     PRINTER, INCLUDING THE VALUES OF ABSCISSA AND ORDINATE AT   PCHEB    81
      C     THE GRID LINES OUTSIDE THE BOTTOM AND LEFT EDGE OF THE GRAPH PCHEB   82
      C                                                                 PCHEB    83
            CALL PLOT4 (LABLE,NCHAR)                                    PCHEB    84
      C                                                                 PCHEB    85
            RETURN                                                      PCHEB    86
      C                                                                 PCHEB    87
      C                                                                 PCHEB    88
       9000 FORMAT(1H1,32X,64HBODY SECTIONAL AREA CURVE AND STRUT WATERPLANE O PCHEB 89
           1UTLINE CURVE FOR//25X,8A10)                                 PCHEB    90
            END                                                         PCHEB    91
```

```
      SUBROUTINE PLOT1 (NSCALE,A,B,C,D,E,F)                           PLOT1      2
C                                                                     PLOT1      3
C     SETUP SPACING AND DETERMINE THE AXIS VALUES                     PLOT1      4
C                                                                     PLOT1      5
C                                                                     PLOT1      6
      INTEGER A,B,C,D                                                 PLOT1      7
      LOGICAL V,H                                                     PLOT1      8
C                                                                     PLOT1      9
      DIMENSION NSCALE(4)                                             PLOT1     10
C                                                                     PLOT1     11
      COMMON/XRPLOTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H PLOT1     12
C                                                                     PLOT1     13
      I=NSCALE(1)                                                     PLOT1     14
      J=NSCALE(2)                                                     PLOT1     15
      K=NSCALE(3)                                                     PLOT1     16
      L=NSCALE(4)                                                     PLOT1     17
      NHL=A                                                           PLOT1     18
      NSBH=B-1                                                        PLOT1     19
      NVL=C                                                           PLOT1     20
      NSBV=D-1                                                        PLOT1     21
      HCHAR=E                                                         PLOT1     22
      VCHAR=F                                                         PLOT1     23
      RETURN                                                          PLOT1     24
      END                                                             PLOT1     25
```

```
      SUBROUTINE PLOT2(XMAX,XMIN,YMAX,YMIN,NSCL1)                      PLOT2      2
C                                                                      PLOT2      3
C     EXAMINE THE MINIMUM AND MAXIMUM VALUES OF ABSCISSA AND ORDINATE  PLOT2      4
C     AND ESTABLISH AN INTERNAL FORMULA FOR COMPUTING THE LOCATION     PLOT2      5
C     IN THE IMAGE REGION CORRESPONDING TO THE POINT TO BE PLOTTED     PLOT2      6
C                                                                      PLOT2      7
C                                                                      PLOT2      8
      REAL LIN                                                         PLOT2      9
      INTEGER VCT,VLCT,HCT,HLCT                                        PLOT2     10
      LOGICAL NSCL1,V,H                                                PLOT2     11
C                                                                      PLOT2     12
      DIMENSION LIN(110)                                               PLOT2     13
C                                                                      PLOT2     14
      COMMON/XRPLOTF/XL,XH,YL,YH,XI,YI,XMOV,YMOV                       PLOT2     15
      COMMON/XRPLOTG/GRAF(11,204)                                      PLOT2     16
      COMMON/XRPLOTQ/II,JJ,KK,LL,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,IX,IY,V,H PLOT2   17
C                                                                      PLOT2     18
      IF(NSCL1) GOTO 1                                                 PLOT2     19
         II=0                                                          PLOT2     20
         JJ=3                                                          PLOT2     21
         KK=0                                                          PLOT2     22
         LL=3                                                          PLOT2     23
         NHL=6                                                         PLOT2     24
         NVL=11                                                        PLOT2     25
         NSBH=9                                                        PLOT2     26
         NSBV=9                                                        PLOT2     27
         HCHAR=1L-                                                     PLOT2     28
         VCHAR=1LI                                                     PLOT2     29
    1 CONTINUE                                                         PLOT2     30
      HLCT=0                                                           PLOT2     31
      HCT=NSBH                                                         PLOT2     32
      DO 2 I=1,204                                                     PLOT2     33
         IF(HLCT.EQ.NHL) GOTO 3                                        PLOT2     34
         IY=I                                                          PLOT2     35
         VLCT=0                                                        PLOT2     36
         VCT=NSBV                                                      PLOT2     37
         DO 4 J=1,110                                                  PLOT2     38
            IF(VLCT.EQ.NVL) GO TO 5                                    PLOT2     39
            IX=J                                                       PLOT2     40
            IF(VCT.EQ.NSBV) GOTO 6                                     PLOT2     41
              LIN(J)=1L                                                PLOT2     42
            GOTO 40                                                    PLOT2     43
    6       CONTINUE                                                   PLOT2     44
               LIN(J)=VCHAR                                            PLOT2     45
               VCT=-1                                                  PLOT2     46
               VLCT=VLCT+1                                             PLOT2     47
   40       CONTINUE                                                   PLOT2     48
            VCT=VCT+1                                                  PLOT2     49
    4    CONTINUE                                                      PLOT2     50
    5    CONTINUE                                                      PLOT2     51
         VLCT=0                                                        PLOT2     52
         IF(HCT.NE.NSBH) GOTO 7                                        PLOT2     53
            HCT=-1                                                     PLOT2     54
            HLCT=HLCT+1                                                PLOT2     55
            DO 8 J=1,IX                                                PLOT2     56
               LIN(J)=HCHAR                                            PLOT2     57
    8       CONTINUE                                                   PLOT2     58
```

```
       7    CONTINUE                                                 PLOT2       59
            HCT=HCT+1                                                PLOT2       60
            IQ=IX+1                                                  PLOT2       61
            DO 10 J=IQ,110                                           PLOT2       62
               LIN(J)=1L                                             PLOT2       63
      10    CONTINUE                                                 PLOT2       64
            ENCODE(110,9,GRAF(1,I)) (LIN(J),J=1,110)                 PLOT2       65
       2 CONTINUE                                                    PLOT2       66
       3 CONTINUE                                                    PLOT2       67
         XL=XMIN                                                     PLOT2       68
         XH=XMAX                                                     PLOT2       69
         YL=YMIN                                                     PLOT2       70
         YH=YMAX                                                     PLOT2       71
         XI=(XH-XL)/FLOAT(IX-1)                                      PLOT2       72
         YI=(YH-YL)/FLOAT(IY-1)                                      PLOT2       73
         XMOV=1.0-XL/XI                                              PLOT2       74
         YMOV=1.0-YL/YI                                              PLOT2       75
         V=.TRUE.                                                    PLOT2       76
         H=.TRUE.                                                    PLOT2       77
         RETURN                                                      PLOT2       78
       9 FORMAT(110A1)                                               PLOT2       79
         END                                                         PLOT2       80
```

```
      SUBROUTINE PLOT3 (PCHAR,X,Y,SDATA,FDATA,DDATA)                    PLOT3       2
C                                                                       PLOT3       3
C     ASSIGN AN ALPHA-CHARACTER TO EACH POINT WHICH WILL BE PLOTTED     PLOT3       4
C                                                                       PLOT3       5
C                                                                       PLOT3       6
      REAL LIN                                                          PLOT3       7
      INTEGER ROW, COL                                                  PLOT3       8
      INTEGER SDATA, FDATA, DDATA                                       PLOT3       9
C                                                                       PLOT3      10
      DIMENSION LIN(110)                                                PLOT3      11
      DIMENSION X(FDATA), Y(FDATA)                                      PLOT3      12
C                                                                       PLOT3      13
      COMMON/XRPLOTF/XL,XH,YL,YH,XI,YI,XMOV,YMOV                        PLOT3      14
      COMMON/XRPLOTG/GRAF(11,204)                                       PLOT3      15
C                                                                       PLOT3      16
      DO 18 I = SDATA, FDATA, DDATA                                     PLOT3      17
         IF (Y(I).LT.YL.OR.Y(I).GT.YH) GOTO 16                          PLOT3      18
            ROW=IFIX(Y(I)/YI+YMOV)                                      PLOT3      19
            IF(ROW.LT.1.OR.ROW.GT.204) GOTO 14                          PLOT3      20
               DECODE(110,2,GRAF(1,ROW)) (LIN(J),J=1,110)               PLOT3      21
               IF(X(I).LT.XL.OR.X(I).GT.XH) GOTO 12                     PLOT3      22
                  COL=IFIX(X(I)/XI+XMOV)                                PLOT3      23
                  IF(COL.LT.1.OR.COL.GT.110) GOTO 10                    PLOT3      24
                     LIN(COL)=PCHAR                                     PLOT3      25
                     ENCODE(110,2,GRAF(1,ROW)) (LIN(J),J=1,110)         PLOT3      26
   10             CONTINUE                                              PLOT3      27
   12          CONTINUE                                                 PLOT3      28
   14       CONTINUE                                                    PLOT3      29
   16    CONTINUE                                                       PLOT3      30
   18 CONTINUE                                                          PLOT3      31
      RETURN                                                            PLOT3      32
    2 FORMAT(110A1)                                                     PLOT3      33
      END                                                               PLOT3      34
```

```
            SUBROUTINE PLOT4(MCHAR,NCHAR)                                     PLOT4     2
      C                                                                       PLOT4     3
      C     PRINT THE IMAGE OF THE COMPLETED GRAPH ON THE PRINTER,            PLOT4     4
      C     INCLUDING THE VALUES OF THE ABSCISSA AND ORDINATE AT THE          PLOT4     5
      C     GRID LINES OUTSIDE THE BOTTOM AND LEFT EDGES OF THE GRAPH         PLOT4     6
      C                                                                       PLOT4     7
      C                                                                       PLOT4     8
            INTEGER SA, SB, SC                                                PLOT4     9
            INTEGER CT,PDQ                                                    PLOT4    10
            LOGICAL V,H                                                       PLOT4    11
      C                                                                       PLOT4    12
            DIMENSION HNUM(15), MCHAR(3), LCHAR(30), VGFMT(3), HLFMT(3)       PLOT4    13
      C                                                                       PLOT4    14
            COMMON/XRPLOTF/XL,XH,YL,YH,XI,YI,XMOV,YMOV                        PLOT4    15
            COMMON/XRPLOTG/GRAF(11,204)                                       PLOT4    16
            COMMON/XRPLOTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H  PLOT4    17
      C                                                                       PLOT4    18
            SA=NSBV                                                           PLOT4    19
            SC=NCHAR                                                          PLOT4    20
            SB=NVL                                                            PLOT4    21
            KQ=1                                                              PLOT4    22
            CALL QPLOTZ5(PDQ)                                                 PLOT4    23
            ENCODE(26,2,VGFMT(1))J                                            PLOT4    24
            IF(NSBV.LE.0) NSBV=10                                             PLOT4    25
          4 CONTINUE                                                          PLOT4    26
            IF(NSBV.GT. 10) GO TO 3                                           PLOT4    27
               NSBV=NSBV*2                                                    PLOT4    28
               XVL=NVL                                                        PLOT4    29
               NVL=XVL/2.0+.5                                                 PLOT4    30
               KQ=KQ+1                                                        PLOT4    31
               GO TO 4                                                        PLOT4    32
          3 CONTINUE                                                          PLOT4    33
            NNN=NVL-1                                                         PLOT4    34
            MMM=NSBV-8                                                        PLOT4    35
            ENCODE(25,5,HLFMT(1)) PDQ,L,NNN,MMM,L                             PLOT4    36
            IF (NCHAR.NE.0) GOTO 14                                           PLOT4    37
               NCHAR=1                                                        PLOT4    38
               LCHAR(1)=1L                                                    PLOT4    39
            GO TO 15                                                          PLOT4    40
         14 CONTINUE                                                          PLOT4    41
               DECODE(10,17,MCHAR(1)) (LCHAR(IJ),IJ=1,10)                     PLOT4    42
               DECODE(10,17,MCHAR(2)) (LCHAR(IJ),IJ=11,20)                    PLOT4    43
               DECODE(10,17,MCHAR(3)) (LCHAR(IJ),IJ=21,30)                    PLOT4    44
         15 CONTINUE                                                          PLOT4    45
            MCT=0                                                             PLOT4    46
            NSPC=(ISY-NCHAR)/2                                                PLOT4    47
            LCT=0                                                             PLOT4    48
            CT=NSBH                                                           PLOT4    49
            DO 6 MMNX=1,ISY                                                   PLOT4    50
               N=ISY-MMNX + 1                                                 PLOT4    51
               IF (MMNX.LE.NSPC.OR.LCT.GT.NCHAR) GO TO 7                      PLOT4    52
                  LCT=LCT+1                                                   PLOT4    53
                  LCR=LCHAR(LCT)                                              PLOT4    54
               GO TO 8                                                        PLOT4    55
          7     CONTINUE                                                      PLOT4    56
                  LCR=1L                                                      PLOT4    57
          8     CONTINUE                                                      PLOT4    58
```

```
         IF(CT.EQ.NSBH.AND.H) GO TO 9                                      PLOT4      59
            PRINT 1,LCR,(GRAF(II,N),II=1,11)                               PLOT4      60
         GO TO 13                                                          PLOT4      61
    9    CONTINUE                                                          PLOT4      62
            CT=-1                                                          PLOT4      63
            QNN=(YH-MCT*YI)*10.**I                                         PLOT4      64
            PRINT VGFMT,LCR,QNN,(GRAF(II,N),II=1,11)                       PLOT4      65
            MCT=MCT+NSBH+1                                                 PLOT4      66
   13    CONTINUE                                                          PLOT4      67
         CT=CT+1                                                           PLOT4      68
    6 CONTINUE                                                             PLOT4      69
      MCT=0                                                                PLOT4      70
      IF(.NOT.V) GO TO 12                                                  PLOT4      71
         DO 10 N=1,NVL                                                     PLOT4      72
            HNUM(N)=(XL+MCT*XI)*10.**K                                     PLOT4      73
            MCT=MCT+NSBV+KQ                                                PLOT4      74
   10    CONTINUE                                                          PLOT4      75
         PRINT 11                                                          PLOT4      76
         PRINT HLFMT,(HNUM(N),N=1,NVL)                                     PLOT4      77
   12 CONTINUE                                                             PLOT4      78
      NSBV=SA                                                              PLOT4      79
      NVL=SB                                                               PLOT4      80
      NCHAR=SC                                                             PLOT4      81
      RETURN                                                               PLOT4      82
    1 FORMAT(1H,A1,16X,11A10)                                              PLOT4      83
    2 FORMAT (15H(1H ,A1,3X,F10.,I1,10H,3X,11A10))                         PLOT4      84
    5 FORMAT(1H(,I2,6HX,F10.,I1,1H,,I2,1H(,I2,6HX,F10.,I1,2H)))            PLOT4      85
   11 FORMAT (1H0)                                                         PLOT4      86
   17 FORMAT(10A1)                                                         PLOT4      87
      END                                                                  PLOT4      88
```

```
      SUBROUTINE QPLOTZ5(PDQ)                                QPLOTZ   2
C                                                            QPLOTZ   3
      INTEGER A,B,C,D,PDQ                                    QPLOTZ   4
      LOGICAL V,H                                            QPLOTZ   5
C                                                            QPLOTZ   6
      COMMON/XRPLOTF/XL,XH,YL,YH,XI,YI,XM,YM                 QPLOTZ   7
      COMMON/XRPLOTQ/II,JJ,KK,LL,A,B,C,D,E,F,M,N,V,H         QPLOTZ   8
C                                                            QPLOTZ   9
      X=ABS(XL)                                              QPLOTZ  10
      IF(X.LT.XH) X=XH                                       QPLOTZ  11
      K=0                                                    QPLOTZ  12
      DO 1 I=1,11                                            QPLOTZ  13
         XJ=10**(I-1)                                        QPLOTZ  14
         IF(X.GT.XJ) GOTO 2                                  QPLOTZ  15
         K=K+1                                               QPLOTZ  16
    1 CONTINUE                                               QPLOTZ  17
    2 CONTINUE                                               QPLOTZ  18
      PDQ=1                                                  QPLOTZ  19
      IF(LL.EQ.0) PDQ=0                                      QPLOTZ  20
      QQQ=PDQ+KK+LL+K                                        QPLOTZ  21
      PDQ=14.0-(10.0-QQQ)/2.0                                QPLOTZ  22
      RETURN                                                 QPLOTZ  23
      END                                                    QPLOTZ  24
```

```
                 FUNCTION FORMDR(VL)                                                FORMDR      2
           C                                                                        FORMDR      3
           C     EVALUATE FORM DRAG COEFFICIENT                                     FORMDR      4
           C                                                                        FORMDR      5
                 DIMENSION X(9), Y(9)                                               FORMDR      6
                 DATA X / 1.325, 1.38, 1.43, 1.48, 1.55, 1.68, 1.9, 2.2, 2.5 /      FORMDR      7
                 DATA Y / .51, .71, .84, .85, .83, .75, .65, .57, .50 /             FORMDR      8
                 IF (VL.GE.1.325.AND.VL.LT.2.5) GO TO 4                             FORMDR      9
                 FORMDR=0.5                                                         FORMDR     10
                 GO TO 5                                                            FORMDR     11
               4 FORMDR = YINTP (VL,X,Y,9)                                          FORMDR     12
               5 RETURN                                                             FORMDR     13
                 END                                                                FORMDR     14
```

```
            FUNCTION YINTP (XA,X,Y,N)                                   YINTP     2
      C                                                                 YINTP     3
      C        INTERPOLATING THROUGH A SET OF DISCRETE DATA             YINTP     4
      C                                                                 YINTP     5
            DIMENSION X(1), Y(1)                                        YINTP     6
          1 DO 10 I=1,N                                                 YINTP     7
            IF (X(I)-XA) 10,10,2                                        YINTP     8
          2 IN=I-2                                                      YINTP     9
            IF(IN) 4,4,6                                                YINTP    10
          4 IN=1                                                        YINTP    11
            GO TO 12                                                    YINTP    12
          6 NN=N-3                                                      YINTP    13
            IF (IN-NN) 12,12,8                                          YINTP    14
          8 IN=NN                                                       YINTP    15
            GO TO 12                                                    YINTP    16
         10 CONTINUE                                                    YINTP    17
         12 IO=IN+3                                                     YINTP    18
            YINTP=0.                                                    YINTP    19
            DO 20 I=IN,IO                                               YINTP    20
            PROD=Y(I)                                                   YINTP    21
            DO 16 J=IN,IO                                               YINTP    22
            IF (I-J) 15,16,15                                           YINTP    23
         15 PROD=PROD*(XA-X(J))/(X(I)-X(J))                             YINTP    24
         16 CONTINUE                                                    YINTP    25
         20 YINTP=YINTP+PROD                                            YINTP    26
         21 RETURN                                                      YINTP    27
            END                                                         YINTP    28
```

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.

6